APPLICATION PROPAGATION ENVIRONMENT (APEX)

APEx Instantiation Services & Specifications

Vision on technology
for a better world

vito.be

DOCUMENT CONTROL

Document Title	APEx Instantiation Services & Specifications
Project	Application Propagation Environment (APEx)
Version	2.1

DOCUMENT APPROVER(S) AND REVIEWER(S)

Version	Name	Company	Action	Date
1.1	Patrick Griffiths Paulo Sacramento	ESA	Approved	26/08/2024
1.4	Patrick Griffiths Paulo Sacramento	ESA	Approved	12/12/2024
2.1	Patrick Griffiths Paulo Sacramento	ESA	Approved	10/07/2025

DOCUMENT VERSION HISTORY

Version	Date	Section	Description	Editor/Reviewer
1.0	20/05/2024	All	Setup initial content	Pedro Gonçalves
1.0	28/05/2024	4, 7, 8, 9	Provided input for VITO services	Bram Janssen
1.0	03/07/2024	3	Provided input for Dashboard	Dan Ormsby
1.1	05/08/2024 – 13/08/2024	All	Updated content based on MS1 RIDs V1.0	Bram Janssen, Jeroen Dries, Pedro Gonçalves Hervé Caumont
1.2	17/10/2024	4,7,8,9	Added roadmap for Project Portal, Product Catalogue, User Forum and Documentation Portal services	Bram Janssen
1.2	21/10/2024	All	Addressed open comments	Hervé Caumont
1.2	28/10/2024	3	Added roadmap for Geospatial Explorer	Dan Ormsby
1.2	28/10/2024	5,6	Added roadmap for User Workspace and IDE services	Hervé Caumont

1.3	12/11/2024	All	Updated content based on feedback	Bram Janssen
1.4	05/12/2024	All	Updated content based on feedback	Bram Janssen Pedro Gonçalves Hervé Caumont
2.0	24/04/2025	All	Added screenshots Added usage and roadmap section	Bram Janssen
2.0	30/04/2025	6	Restructured to describe individual SaaS products	Hervé Caumont
2.0	05/05/2025	2.2.3, 2.2.4	Updated descriptions of key technologies	Zdenek Jurman
2.0	08/05/2025	3	Updated text of Geospatial Explorer	Adam Tweedie
2.1	26/05/2025	Preamble, 2.2, 3,5, 6	Updated content based on MTR RIDs V1.0	Bram Janssen Hervé Caumont

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide the Service Specification for the APEx Instantiation Services.

1.2 Scope

APEx (Application Propagation Environments) aims to offer a comprehensive suite of services designed to support every stage of EO project development and propagation. By providing robust, scalable, and user-friendly tools, APEx will ensure that EO data and applications are accessible, reusable, and impactful, fostering a vibrant and collaborative EO community.

The APEx project will develop and provide a set of services designed to support the sustainable availability, propagation, and practical utilisation of EO data, algorithms, and project results. These services are divided into several categories, each targeting specific aspects of the project lifecycle from development to dissemination. Below is a detailed overview of the different APEx services:

- **Instantiation Services**: designed to create and configure working environments for projects, providing the necessary infrastructure and tools to start and manage EO projects efficiently.
- **Propagation Services**: focus on ensuring that developed algorithms and workflows are effectively scaled, enhanced, and made available for broader use.
- **Support and Operational Services**: ensure the ongoing support, maintenance, and operational efficiency of APEx and its components.
- **Specific Project-Related Services**: tailored to specific projects and their unique requirements, ensuring that the needs of each project are met effectively.

1.3 Document Structure

This Service Specification document is divided into the following Chapters:

- Chapter 1 (current) - Introduction
- Chapter 2 - Overview of the APEx Instantiation Services, the key technologies supporting the services, its value proposition and use cases.
- Chapter 3 - Service Specification for the Dashboard Services
- Chapter 4 - Service Specification for the Project Portal Service
- Chapter 5 - Service Specification for the User Workspace Service
- Chapter 6 - Service Specification for the IDE & GIS Collaborative Services
- Chapter 7 - Service Specification for the Product Catalogue Service
- Chapter 8 - Service Specification for the Documentation Portal Service
- Chapter 9 - Service Specification for the User Forum Service

1.4 Applicable Documents

Ref.	Title and References	Issue
[SOW]	Application Propagation Environment (APEx) (ESA-EOP-SD-SOW-0426)	1.0
[CTR]	**ESA** Contract No. 4000143982/24/I-EB with **VITO NV** for *Application Propagation Environment (APEx)*	1.0

1.5 Reference Documents

Ref.	Title and References	Issue
[PROP]	TAP/22114/231123 Proposal for *Application Propagation Environment* (all Volumes)	1.0

2 OVERVIEW

2.1 APEx Instantiation Services

The APEx Instantiation Services are designed to provide managed, configurable environments that support the collaboration, development, exploration, and visualisation of application project results. These services are crucial for facilitating the effective sharing, maintenance, and utilisation of project outcomes within the Earth Observation (EO) community, ensuring that they remain readily accessible and usable for extended periods.

APEx aims to cater to the diverse and multifaceted needs of the EO community by delivering Software as a Service (SaaS) products. It manages the delivery of single-user or shared work environments within a unified cloud infrastructure, facilitating a wide range of user tasks, including development, hosting, execution, and exploratory analysis of EO applications.

At the core of the APEx Instantiation Services is the ability to manage and deliver these environments effectively. Project-wide community-oriented services, such as the portal, catalogue, documentation portal, and user forum, are managed directly by Kubernetes and typically instantiated once per project. Conversely, single-user-specific workspaces allow the instantiation of services by a single user at any time. These services, including the User Workspace, Interactive Development Environment (IDE), and, in certain cases, dashboards and web applications, are managed by JupyterHub. JupyterHub orchestrates the launching and management of these software deployments, ensuring isolated and customisable environments for individual users.

A key feature of the APEx Instantiation Services is the ability to provide managed, configurable environments. These environments can be tailored to the specific needs of different projects and users, supporting various tasks, including development, hosting, execution, and exploratory analysis of EO applications. The services are particularly aimed at supporting ESA's application projects and the four thematic Stakeholder Engagement Facility (SEF) instances. This flexibility and configurability allow these projects to focus on their primary research objectives without being bogged down by the technical complexities of setting up and maintaining their workspaces.

Another critical component of the APEx Instantiation Services is the seamless integration with the Network of Resources (NoR). The focus is on developing delta instantiation services, which are those that do not currently exist in the NoR. These new services are to be developed and onboarded into the NoR, making them available as payable services. For instantiation service components that already exist, such as the ELLIP IDE, the goal is to integrate these existing services. The technical challenge lies in achieving seamless integration and alignment of business models, ensuring that newly developed and existing services work harmoniously within the APEx platform.

The specific components of the APEx Instantiation Services include:

- **Geospatial Explorer**: Provide a data-driven user interface to display and visualise geospatial and tabular data from a range of supported web services based on a configuration defined by a dashboard administrator.
- **Project Portals**: Creating a project website based on Drupal, working in synergy with other APEx services.
- **User Workspaces**: Offering secure and personalised work environments with data-sharing mechanisms.
- **Interactive Development Environments:** Leveraging Code Server (VS Code in the browser) tailored specifically for EO tasks.
- **Product Catalogues:** Featuring SpatioTemporal Asset Catalog (STAC) catalogues and streamlined data ingestion processes.
- **Documentation Portals:** Supporting customisation and interactive visualisation using the Quarto framework.
- **User Forums:** Providing a community-building platform based on open-source software Discourse.

The APEx Instantiation Services will be essential for ensuring that the results of EO projects are effectively shared and utilised, fostering greater collaboration and innovation within the EO community. By providing robust, scalable, and user-friendly environments, the APEx Instantiation Services help maximise the impact of EO research and applications.

2.2 Key Technologies

The APEx Instantiation Services leverage a combination of advanced technologies to manage and deliver both *single-user* and project-wide, community-oriented work environments, ensuring robust, scalable, and user-centric solutions for the EO community. Central to this operation are Kubernetes and JupyterHub, each playing a crucial role in orchestrating the deployment and management of these environments.

This architectural design ensures that the APEx Instantiation Service remains modular, scalable, and capable of catering to the ever-evolving requirements of EO tasks.

It covers in particular, the following key enabling technologies:

- Keycloak, with fine-grained permission management configured, along with the definition of per-project roles. We present Keycloak further below as part of the "Shared Services".
- Kubernetes Cluster Autoscaler, with fine-tuning for ensuring the performance and stability of the cluster instances.
- Alloy/Mimir time-series metrics, along with the definition of resource limits to protect the stability of the workload of the project.
- Monitoring of the APEx environment with Grafana, managing the full list of Ingress controllers and featuring project monitoring dashboards.

2.2.1 Project-Wide Work Environments

Kubernetes plays a fundamental role in managing shared work environments within the APEx Instantiation Services. Kubernetes is an open-source platform designed for automating the deployment, scaling, and operations of application containers across clusters of hosts. This ensures that applications run in isolated environments, improving reliability and resource efficiency.

Key functionalities of Kubernetes in the context of APEx include:

- **Container Orchestration**: Kubernetes manages the deployment and scaling of containerised applications, ensuring that each service runs efficiently across the cluster. This is particularly important for shared services such as the project portal, product catalogue, documentation portal, and user forum.
- **Load Balancing and Service Discovery**: Kubernetes provides built-in load balancing and service discovery, ensuring that incoming traffic is distributed evenly across instances of an application. This enhances the availability and performance of multi-user-facing services.
- **Automated Rollouts and Rollbacks**: Kubernetes automates the rollout and rollback of application updates, ensuring that new versions of applications can be deployed without downtime. This capability is crucial for maintaining the reliability of multi-user shared services.
- **Resource Management and Monitoring**: Kubernetes offers robust resource management and monitoring tools, allowing administrators to track the health and performance of applications in real time. This ensures that resources are used efficiently and potential issues are addressed proactively.
- **Shared**: Kubernetes enforces security policies and isolates application environments, ensuring that multi-user shared services comply with security and regulatory standards. This includes managing access controls and network policies to protect sensitive data.

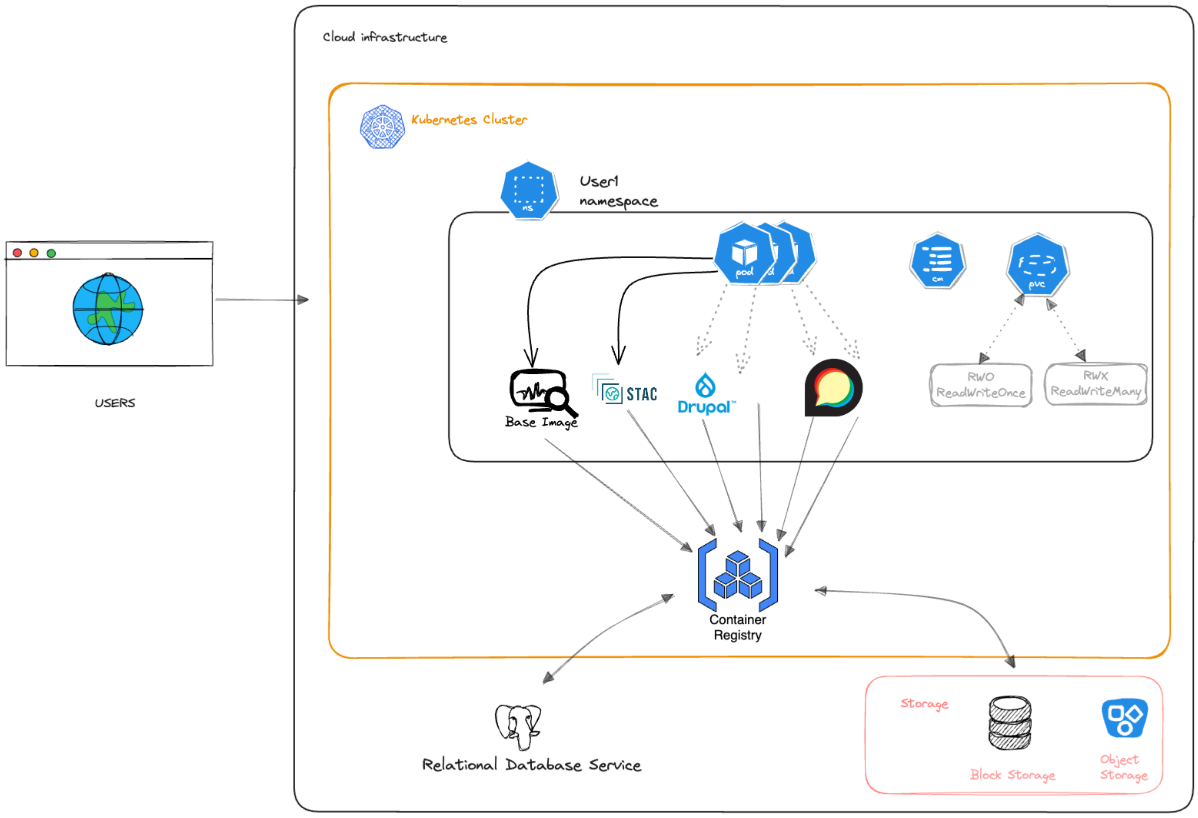


Figure 1. Architecture of Project-Wide Work Environments in APEx Instantiation Services

By utilising Kubernetes, APEx ensures that shared work environments are scalable, reliable, and secure, capable of supporting the diverse needs of the EO community. Figure 1 illustrates the architecture and workflow of Shared Work Environments managed within the APEx Instantiation Services. It shows how user requests are handled within the Kubernetes cluster and how different components interact to provide shared services, highlighting:

- **User Interaction:** Users interact with the system through a web interface of the specific service (e.g. portal, forum) here, they can manage and further customise their environment. Their requests are directed to the Kubernetes cluster, where the required services and environments are managed.
- **Kubernetes Cluster:** The core of the shared work environment is managed within a Kubernetes cluster. This cluster orchestrates the deployment, scaling, and management of containerised applications.
- **Namespaces and Resource Management:** Each user or group of users is allocated a namespace within the Kubernetes cluster. These namespaces ensure resource isolation and management, allowing different users to operate in separate virtual environments. These work environments can be expanded with the use of vcluster, or Virtual Cluster, which can be used to create lightweight, fully functional Kubernetes clusters within a shared Kubernetes environment.
- **Base Image and Container Registry:** Base images contain essential tools and configurations and are used to create user environments. The Container Registry stores these images, which are then pulled to create containers (pods in deployments) for different applications.

- **Applications and Services:** The environment supports various applications and services, such as a STAC Browser (Catalog for managing EO data), Drupal (a content management system for creating and managing project portals), Quarto (for generating the documentation portal and managing its content) and Discourse (a platform for user forums and community discussions). These applications are instantiated from the Container Registry and deployed within the user namespaces.
- **PersistentVolumeClaims (PVCs) and ConfigMaps (CM):** PVCs are created to request storage resources for the user environments. These specify the size and access modes (ReadWriteOnce and ReadWriteMany) required for data persistence. ConfigMaps store configuration data separately from the application's code, ensuring that necessary environment settings are applied to the containers.
- **Data Storage:** The system integrates with both block storage and object storage solutions to provide persistent storage for user data. This ensures that data is securely stored and accessible as needed.
- **Relational Database Service:** A relational database service, such as PostgreSQL, is used to store metadata and user information and manage authentication and user data for various applications.

2.2.2 Single-User Work Environments

Central to the APEx Instantiation Service's operation for single-user work environments is JupyterHub, which orchestrates the launching, scaling, and management of software deployments. JupyterHub is designed to support the JupyterLab application, which showcases its versatility by functioning as a Software-as-a-Service (SaaS) solution, including Code Server (Visual Studio Code on the Cloud) and remote desktop applications, notably QGIS and SNAP. Additionally, JupyterHub facilitates the seamless integration and presentation of various dashboard frameworks. By dynamically provisioning user-dedicated environments, JupyterHub acts as the primary gateway, channelling user requests to the respective isolated application pods, ensuring efficient resource utilisation and streamlined user access.

The APEx Instantiation Service leverages the dynamic capabilities of Kubernetes and JupyterHub to provide personalised and context-aware application experiences. This is achieved through several key features:

- **Namespaces**: Each application pod is deployed in its dedicated namespace, ensuring an organised, secure, and isolated environment. This setup facilitates easy management, monitoring, and logging of individual application instances.
- **Application Contextualization**: This involves dynamically setting up the application pod based on the task at hand. It ensures that the pod has all necessary configurations, from environment variables to specific files (like S3 or Docker configurations), ensuring the application runs optimally.
- **User Profile Management with Kube Spawner**: The Kube Spawner plays a crucial role in contextualising the application pod experience based on user profiles. From managing authentication and authorisation protocols to handling lifecycle events with pre-spawn and post-stop hooks, it ensures a seamless user experience.

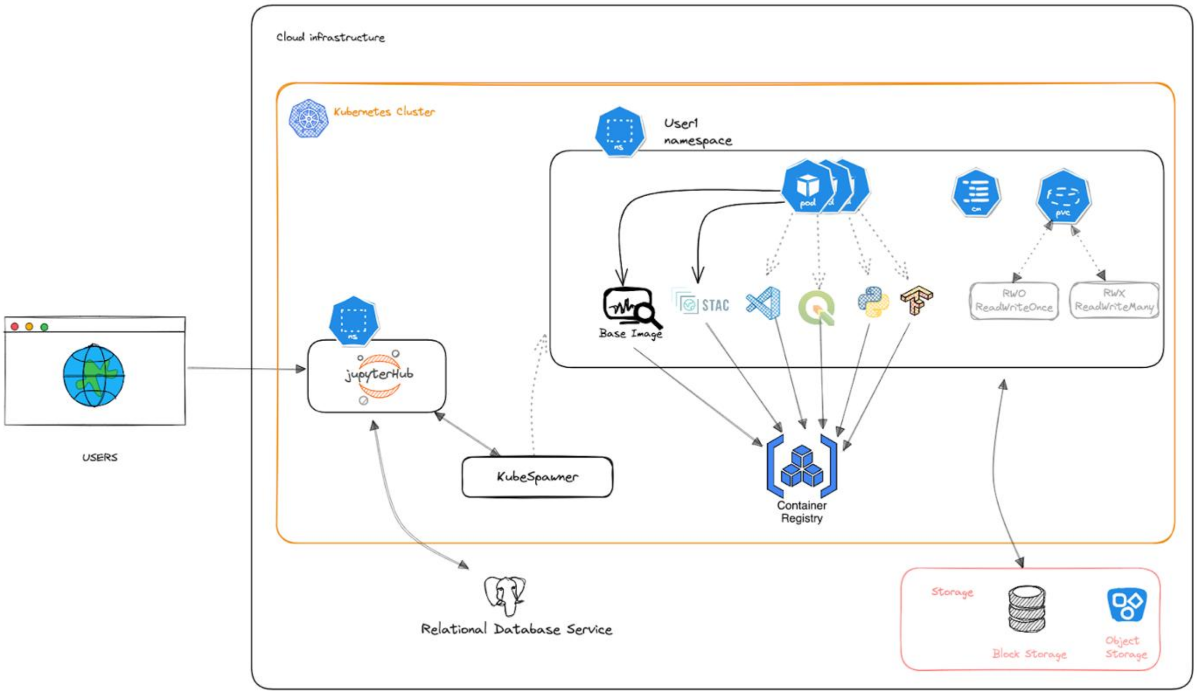


Figure 2. Architecture of Single-User Work Environments in APEx Instantiation Services

Figure 2 illustrates the architecture and workflow of single-user work environments managed by JupyterHub within the APEx Instantiation Services. It highlights the key components and processes involved in provisioning and managing user workspaces:

- **JupyterHub Management:** JupyterHub provides a web interface for authenticated users to instantiate their Application Hub pod instances.
- **KubeSpawner:** Upon server initiation, the KubeSpawner dynamically creates and manages Kubernetes resources, including namespaces, ConfigMaps, PersistentVolumeClaims (PVCs), and application pods.
- **Dedicated Namespace:** Each user receives a dedicated namespace within the Kubernetes cluster for resource isolation and management.
- **ConfigMap:** Stores configuration data such as Conda environment configurations.
- **PersistentVolumeClaims (PVCs):** Request storage resources with specified sizes and access modes (ReadWriteOnce and ReadWriteMany).
- **Pod Creation:** Pods are created using dedicated profiles based on user selection.
- **Database Integration:** A PostgreSQL database stores user information for JupyterHub, managing authentication and user data.
- **Data Management:** Secure storage and retrieval of data facilitated through PVCs and integration with external data sources.
- **Storage Solutions:** Block storage and object storage are provided for persistent data management.

2.2.3 Shared Services for Work Environments

Shared services on APEx's Kubernetes cluster provide a robust foundation that not only accelerates deployment times but also ensures security, observability, and encourages best practices. Shared services empower ICT specialist teams to rapidly deploy and iterate on new deployments with confidence, ensuring that critical infrastructure components are both robust and secure as Instantiation services continue to evolve.

- **Alloy**: Serve as a middleware layer to collect data from sources (container, service or plugin) towards a back-end store. Mimir is used as a back-end store for time-series metrics and Loki for log data. It performs lightweight enrichment (adding metadata such as environment, cluster, or application tags) and passes this on to Grafana. Alloy has native pipelines for leading telemetry signals, such as Prometheus and OpenTelemetry, and databases such as Loki and Pyroscope. This permits logs, metrics, traces, and even mature support for profiling.

- **Cert Manager**: Acts as an automated manager and controller for TLS certificates. Its main responsibilities include requesting, issuing, and renewing certificates from Let's Encrypt, using the Automatic Certificate Management Environment protocol. By automating certificate renewal and deployment, cert-manager minimises human error and avoids downtime associated with expired or misconfigured certificates. Its tight integration with Kubernetes ensures that applications exposed to the public are always served over HTTPS with valid TLS certificates.

- **Grafana**: Plays a central role as the visualisation and user interface layer in a Kubernetes cluster. Sits on top of the observability stack and unifies data from different back ends. Whether it's log streams from Loki, metrics from Mimir, or data/analytics from Alloy, Grafana aggregates these disparate data sources into a single pane of glass. Grafana's broad set of visualisation options—heatmaps, bar charts, graphs, tables, and more—allows users to tailor dashboards to their specific needs.

- **Keycloak**: In APEx, Keycloak serves as the central authentication server for users, services, and workloads. It is tightly integrated with the following identity providers: EGI, ESA EOIAM and Inuits Keycloak. Once the identity is established, authorisation determines the actions that the authenticated user or service is permitted to perform. Built-in integration with OIDC helps many modern web, mobile, and microservices applications to integrate easily with Keycloak. Standardised endpoints and token formats mean less custom code and easier maintenance. APEx Keycloak is configured to dynamically map users and services to a predefined role structure covering granular access levels for each project and work environment. The Keycloak web user interface is also used for self-service management of project users and services by dedicated project admins.

- **Loki**: Grafana Loki is a log aggregation system designed with a philosophy similar to Prometheus. Deploying it on a Kubernetes cluster plays a crucial role in managing, aggregating, and analysing log data across containerised applications and services. It centralises log data, making logs easier to search and correlate with metrics collected from the cluster. This is vital for troubleshooting, performance monitoring, and debugging distributed applications. Loki integrates seamlessly with Grafana, allowing users to visualise and query logs alongside metrics in a single unified dashboard.

- **Mimir**: Grafana Mimir is a specialised, horizontally scalable time series database optimised for metrics storage and querying. Mimir is designed to efficiently ingest, store, and query large volumes of Prometheus metrics. Within a Kubernetes deployment, this means it can serve as a central repository of application, node, and system metrics, ensuring that monitoring data is retained over long periods for analysis and troubleshooting. Mimir employs optimised data structures along with techniques like index caching and chunk storage to improve both ingestion rates and query responsiveness. This efficiency is key when dealing with high cardinality and large volumes of time-series data typical in microservices environments.

- **Vault**: Deployed as a secret engine, Vault provides secure storage and controlled access to sensitive data like API keys, credentials, and certificates. Vault deployment in Kubernetes allows containerised applications to obtain secrets in a controlled manner using Kubernetes-authenticated methods. It is tightly integrated with Kubernetes' Role Based Access Control and service accounts, ensuring that only authorised pods or services are given the credentials they need.

2.2.4 Internal Key Kubernetes Services

Serving as the foundation for hosted Work Environments, the underlying Kubernetes services are essential for robust, resilient and secure workload deployments. Kubernetes's control plane components work together to ensure the cluster can automatically adapt to increasing workloads and recover autonomously from failures. This level of orchestration and self-healing allows future workloads to scale without major architectural changes.

- **Cluster Autoscaler:** Built-in component that automatically adjusts the size of your cluster's node pool based on the current workload demands. When there are pending pods that cannot be scheduled due to insufficient resources, the autoscaler requests additional nodes. This helps you handle sudden spikes in workloads without manual intervention. Conversely, if nodes run at low utilisation for a configurable period, the autoscaler will evaluate whether it's safe to remove them, ensuring that any running pods are either terminated gracefully or rescheduled on other nodes.

- **CoreDNS**: Handles automatic service discovery by mapping Kubernetes service names to IP addresses. This enables pods and services to communicate using DNS names instead of IP addresses. CoreDNS is installed as the cluster's primary DNS service, integrated as a plugin into the Kubernetes control plane.

- **Everest Container Storage:** By conforming to the CSI standards, Everest registers itself as a storage provider, allowing Kubernetes to call on its capabilities to provision and manage persistent volumes (PVs) and persistent volume claims (PVCs). The plugin is designed to support dynamic provisioning, meaning that storage volumes are automatically created on demand when an application requests them via a PVC. It can scale to meet the needs of growing clusters or workloads, thereby accommodating both high-performance and high-capacity requirements.

- **Istio:** Service mesh designed to abstract and manage the communication between microservices. Istio supports advanced traffic routing capabilities, such as canary deployments, A/B testing, and blue-green releases, through granular control over request flows. Istio simplifies the process of securing service-to-service communication by enabling automatic encryption and strong identity-based authentication. Backed by a strong open-source community, Istio has been battle-tested in production environments at scale, continuously iterating and improving based on real-world needs.

- **Metrics Server:** Providing key insights into CPU and memory usage at both the node and pod levels. The Metrics Server aggregates resource usage data from the kubelet's built-in monitoring (for example, via cAdvisor) across all nodes in the cluster. Unlike traditional, full-fledged monitoring systems that store long-term metrics, the Metrics Server focuses on providing a real-time snapshot of resource utilisation, making it ideal for immediate operational needs.

2.3 Value Proposition

The APEx Instantiation Services offer a transformative approach to managing EO projects, providing significant value to researchers, developers, and stakeholders within the EO community. By leveraging advanced technologies and user-centric design principles, APEx ensures that project results are effectively shared, maintained, and utilised, driving innovation and maximising impact. The key value propositions of the APEx Instantiation Services include:

- **Enhanced Collaboration and Development:** APEx Instantiation Services create managed, configurable environments that support collaboration and development. These environments are tailored to the specific needs of projects, facilitating seamless teamwork and efficient project execution. The APEx services support a wide range of tasks, from development and hosting to execution and exploratory analysis, ensuring that users can focus on their core research objectives without being hindered by technical complexities.

- **Scalability and Flexibility:** By utilising Kubernetes and JupyterHub, APEx provides a robust and scalable infrastructure that can handle varying workloads and user demands. Shared services, such as project portals and documentation, are managed directly by Kubernetes, ensuring high availability and efficient resource utilisation. Single-user environments, including user workspaces and interactive development environments, are managed by JupyterHub, providing isolated and customisable workspaces for individual users. This scalability and flexibility are crucial for accommodating the diverse and evolving needs of the EO community.

- **Seamless Integration and Interoperability:** APEx Instantiation Services are designed to seamlessly integrate all components, including both project-wide and single-user environments, to provide a cohesive and unified user experience. Public services managed by Kubernetes, such as the project portal, product catalogue, documentation portal, and user forum, work harmoniously with single-user environments managed by JupyterHub. This integration ensures that users can move effortlessly between different services and environments, whether they are accessing a public portal or working in their personalised workspace. Moreover, APEx Instantiation Services are designed to seamlessly integrate with the Network of Resources (NoR), making them available as payable services. This integration ensures that newly developed services can work with existing NoR platform services, forming unified and configurable environments for collaboration and development. The use of open standards and APIs facilitates interoperability in both cases, allowing users to easily connect with other tools and data sources, thereby enhancing the overall utility and accessibility of APEx.

- **Comprehensive Support and User Engagement:** APEx provides support mechanisms to enhance user satisfaction and productivity. Personalised user workspaces, interactive development environments, and documentation portals ensure that users have access to the tools and resources they need. Additionally, the user forum facilitates community engagement and knowledge sharing, enabling users to seek help, share experiences, and collaborate on solutions. This comprehensive support framework helps users navigate the APEx services effectively and maximises their productivity.

- **Advanced Data Visualization and Analytics:** APEx offers tools for data visualisation and analytics, enabling users to interpret and communicate EO data insights effectively. Web-based visualisations support both raster, vector and tabular data, while dashboards provide interactive and real-time data visualisations. The interactive development environments are equipped with tools for data analysis, statistical computation, and supporting complex analytical workflows. These capabilities help users derive meaningful insights from EO data, driving informed decision-making and impactful research.

- **Efficient Data Processing Workflows:** APEx streamlines data processing workflows by providing secure and scalable environments for data ingestion, processing, and analysis. The user workspaces and interactive development environments are designed to handle EO data efficiently, abstracting data management complexities and simplifying the execution of complex workflows. This efficiency in data processing is critical for enabling timely and accurate analyses, which are essential for various EO applications.

- **Accessibility and Maintenance:** One of the key value propositions of the APEx Instantiation Services is the accessibility and maintenance of project results. By providing managed environments and support structures, APEx ensures that EO project outcomes remain accessible and usable throughout the project lifecycle.

2.4 Customisation

The APEx Instantiation Services are designed to adapt to the unique needs of each project. Except for the User Workspace and IDE service, which are offered as a combined package, all other services can be requested and instantiated independently. This modular approach allows projects to select only the tools that align with their specific requirements. Each instantiated service is accessible via a public URL, enabling seamless integration into existing project components.

In addition to the flexibility of service selection, each service can be customised to reflect the project's branding and aesthetics. The degree of customisation depends on the capabilities of the underlying software, typically focusing on general visual elements such as logos and colour schemes.

Currently, customisation is performed during the instantiation process in close collaboration with the project team. As the APEx project evolves, additional mechanisms to streamline and enhance service customisation will be explored. For projects requiring more detailed adjustments, APEx will provide access to the software’s technical documentation, empowering teams to further tailor the services to their needs.

2.5 Creating a collaborative project environment

As shown in section 2.8, APEx offers a flexible suite of tools that can be instantiated independently or integrated to form a cohesive, collaborative project environment. Projects are encouraged to tailor the setup to their specific needs, combining the various APEx tools in ways that enhance collaboration, accessibility, and visibility.

Projects can interconnect their instantiated services in several ways:

- **Integrated Navigation**: If a project already has a public-facing website or uses the APEx Project Portal, links to the instantiated tools, such as the Geospatial Explorer, Product Catalogue or JupyterLab, can be embedded directly into the site's navigation structure. This provides a unified access point for project stakeholders and visitors to the project website.

- **Public and Private Tool Segmentation**: Projects can choose to expose only selected tools to the public while restricting others to registered team members. For instance, the Geospatial Explorer might be made publicly available for data discovery, while the collaborative JupyterLab remains limited to project members.

By thoughtfully providing access to the APEx instantiated services, projects can create a rich, interactive, and secure environment that supports ongoing collaboration, knowledge sharing, and engagement with both internal and external audiences.

2.6 Procedure

To request the instantiation of one or more components for a project, the project team should contact the APEx team. An exploratory meeting will be arranged where APEx provides a demonstration of its services through a dedicated demo environment. This allows the project to explore the capabilities and features of all APEx instantiation services.

Based on the project's specific needs, APEx will instantiate the required components and grant administrator access to the project team. Since all instantiation services are modular and can function independently, projects can select only the components they require.

To adapt and configure these services to their needs, project members will need to create an account within the APEx environment. Currently, the access configuration of administrators and members is configured based on a request basis.

As part of the continuous improvement of the APEx project, the team is actively exploring ways to enhance the request procedure. Improvements, such as enhanced user management features, enabling project administrators to manage access for their team members directly, were already identified. Additionally, APEx will onboard its instantiation services onto the NoR platform, providing projects with a streamlined and structured way to request access to APEx instantiation services.

2.7 Accessibility

Each instantiated component is assigned a dedicated URL following the convention:
https://<tool>.<project>.apex.esa.int.

The only exception to this rule is the project portal, which is hosted at:
https://<project>.apex.esa.int.

This URL assignment process is fully automated by APEx, including the generation of the necessary security certificates. For example, if a project named *project-a* requests the instantiation of a user forum and a project portal, the following URLs will be created:

- https://project-a.apex.esa.int – the project portal
- https://forum.project-a.apex.esa.int – the project's user forum

This granular URL assignment ensures a clear separation between the different instantiated components, making them easy to manage and access. Additionally, as these URLs are publicly accessible, projects can seamlessly integrate the components into their existing websites by linking to them directly.

In addition to the default URLs provided during component instantiation, APEx also supports the use of custom domains owned by the requesting project. Configuring custom domains requires additional setup, which is handled upon request.

2.8 Use Cases

The APEx Instantiation Services are designed to support a wide range of activities within the EO community, from project promotion and user support to data visualisation, data analytics, and data processing. By providing managed, configurable environments, these services facilitate the effective sharing, maintenance, and utilisation of project results, ensuring they remain readily accessible and usable for extended periods.

This section provides an initial outline of the primary use cases for APEx Instantiation Services, demonstrating how they address the specific needs and challenges faced by users and project managers in the EO domain. Each use case highlights the relevant components of the APEx services, detailing how they work together to provide robust solutions for various tasks.

From promoting EO projects through interactive portals to supporting users with personalised workspaces and comprehensive documentation, the APEx Instantiation Services are designed to enhance productivity and foster collaboration. Data visualisation tools enable users to interpret and communicate insights effectively, while advanced data analytics environments provide the computational power and tools necessary for complex analyses. Data processing workflows are streamlined through secure, scalable environments, ensuring that users can handle large volumes of EO data efficiently.

It is important to highlight that the APEx Project Tools can be combined in different ways to create a flexible collaborative project environment. This gives projects the flexibility to adapt the tools based on their needs. For example, they can choose to make the project portal as their main entry point for both project members and their community, linking to the different APEx tools from within the main website. Alternatively, projects can decide to only keep the project portal for public communication and dissemination, while having the user workspaces only visible for the project members. In essence, the APEx Project Tools provide an a la carte approach for setting up and configuring your project environment based on your needs.

The following use cases provide detailed scenarios of how the APEx Instantiation Services can be leveraged to meet the diverse needs of the EO community, maximising the impact of EO research and applications.

2.8.1 Project Promotion

Use Case Description

Promote EO projects by providing a centralised, customisable portal that showcases project outcomes, visualisations, and data products. This use case supports the dissemination and accessibility of project results to a broader audience, including stakeholders, policymakers, and the general public.

Key Components

- Project Portal: Provides dynamic instantiation using Drupal, allowing project managers to create and maintain a visually appealing and informative site.
- Geospatial Explorer: Create an online dashboard to present key metrics and visualisations.
- Documentation Portal: Customizable portals using the Quarto framework for detailed project documentation and interactive visualisations.
- User Forum: Supports community engagement and discussions around the project outcomes.

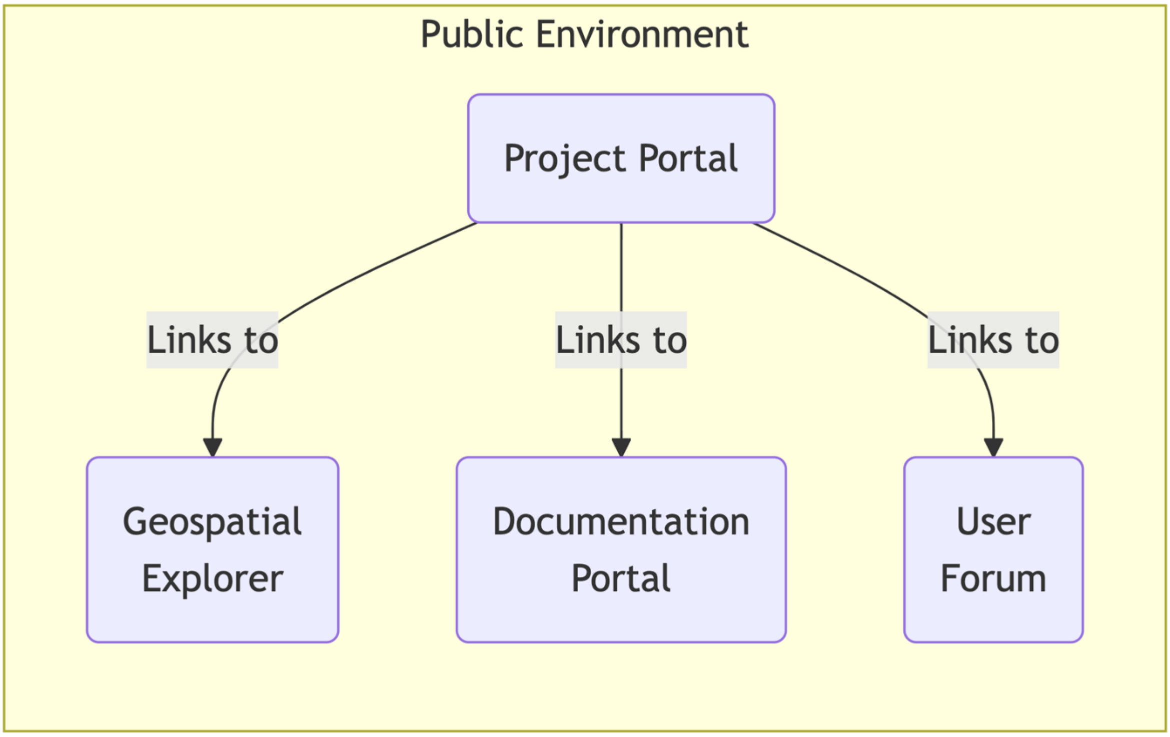


Figure 3. APEx Project Tools – Project Promotion

Expected Benefits

- Increased visibility and impact of EO projects.
- Enhanced stakeholder engagement through accessible and interactive content.
- Long-term maintenance and accessibility of project results.

2.8.2 User Support

Use Case Description

Provide comprehensive support to users, enabling them to interact with EO data, tools, and services effectively. This use case focuses on offering personalised and collaborative environments where the project PI or their team can seek help, share knowledge, and collaborate on EO applications.

Key Components

- Project Portal: Provides dynamic instantiation using Drupal, allowing project managers to create and maintain a visually appealing and informative site.
- User Workspace: Secure and personalised environments allow users to store, share, and manage their data and work.
- Interactive Development Environment (IDE): Leveraging VS Code Server, these environments are tailored for EO tasks and provide development tools.
- User Forum: An on-demand service based on Discourse, facilitating user discussions, support queries, and community-driven solutions.
- Documentation Portal: Comprehensive documentation and tutorial materials to assist users in navigating and utilising the services.

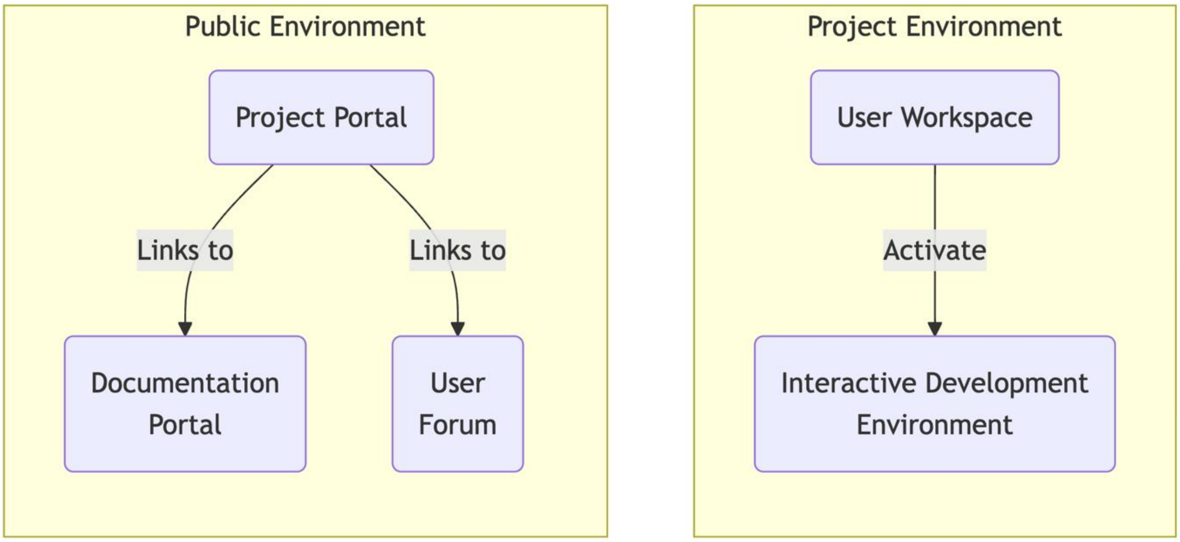


Figure 4. APEx Project Tools - User Support

Expected Benefits

- Improved user satisfaction and productivity through robust support mechanisms.
- Enhanced community engagement and knowledge sharing.
- Efficient troubleshooting and problem-solving through community forums.

2.8.3 Data Visualization

Use Case Description

Enable users to visualise EO data in meaningful and interactive ways. This use case supports both public and project-specific visualisations, helping users to interpret and communicate data insights effectively.

Key Components

- Geospatial Explorer: Functional enhancements to provide interactive and real-time data visualisations, supporting both raster and tabular data visualisations.
- Project Portal: Integration of visualisations into project portals to showcase data insights.
- Documentation Portal: Interactive visualisations within documentation to explain data findings and methodologies.

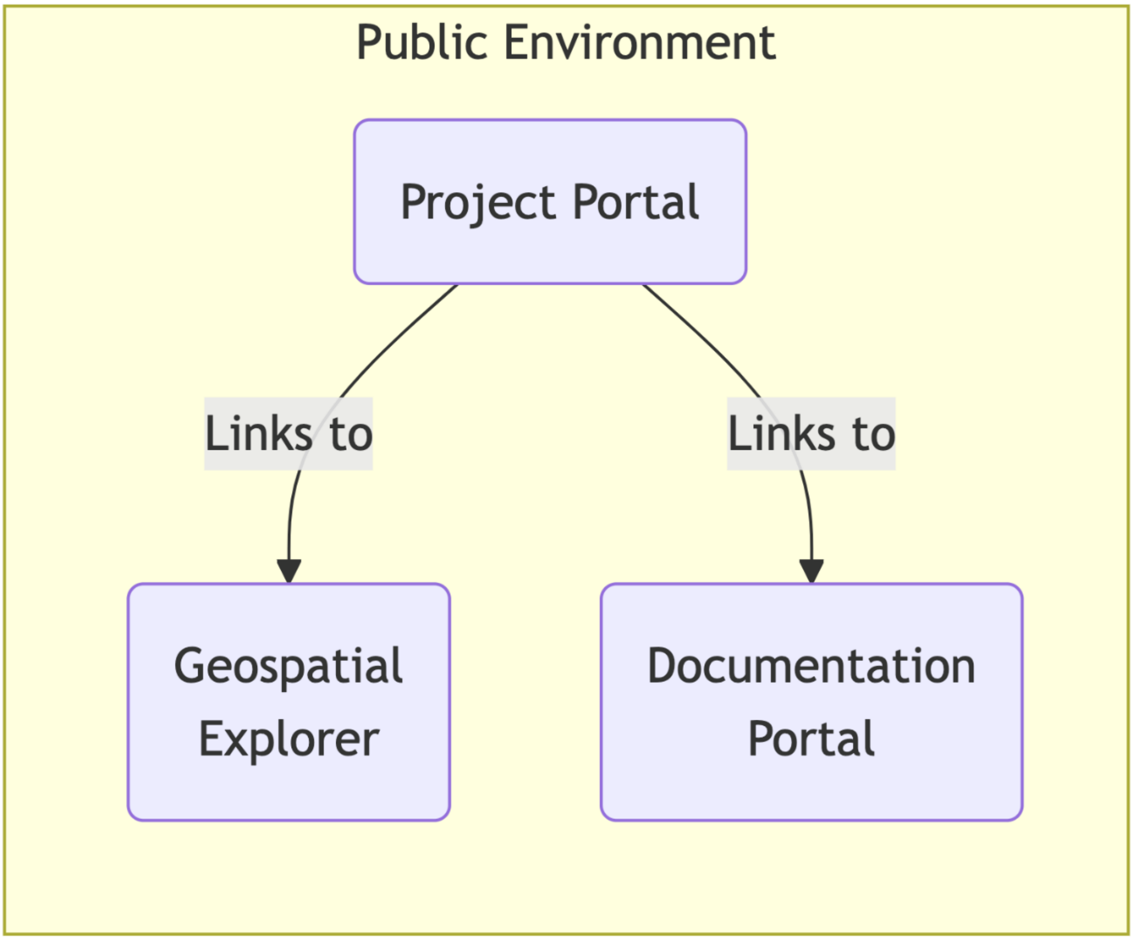


Figure 5. APEx Project Tools - Data Visualisation

Expected Benefits

- Enhanced understanding and interpretation of EO data.
- Improved data communication and storytelling capabilities.
- Increased engagement through interactive and visually appealing content.

2.8.4 Data Analytics

Use Case Description

Support advanced data analytics workflows, enabling users to analyse EO data efficiently. This use case focuses on providing powerful tools and environments for data processing and statistical analysis.

Key Components

- Interactive Development Environment (IDE): Equipped with tools for data analysis, statistical computation, and machine learning, leveraging the capabilities of VS Code Server.
- User Workspace: Secure and scalable environments for conducting extensive data analysis.
- Project Portal: Integration of visualisations into project portals to showcase data insights.
- Product Catalogue: A STAC catalogue providing easy access to EO datasets and streamlining the data ingestion process.
- Geospatial Explorer: Integration with analytics tools to provide real-time visualisations of analytical results.

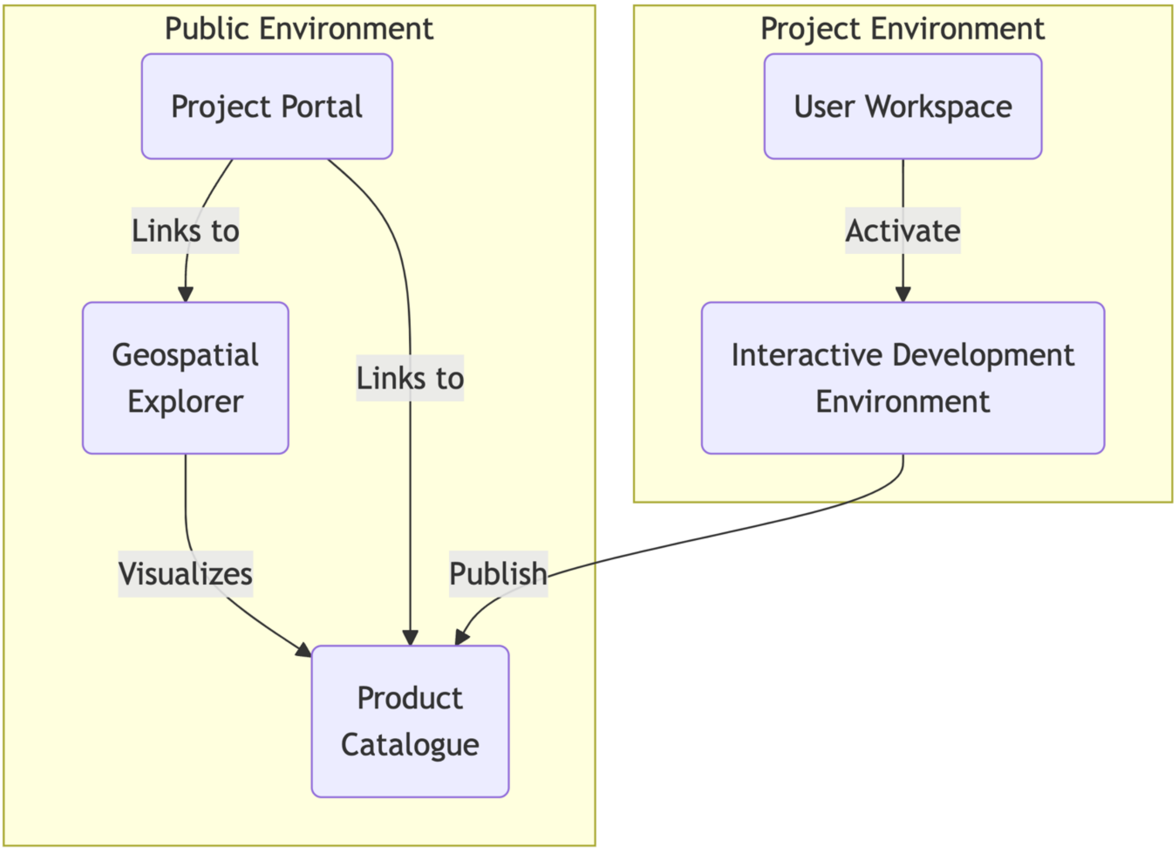


Figure 6. APEx Project Tools - Data Analytics

Expected Benefits

- Enhanced analytical capabilities and data-driven insights.
- Efficient processing and analysis of large EO datasets.
- Streamlined workflows from data ingestion to analysis and visualisation.

2.8.5 Data Processing

Use Case Description

Facilitate the processing of EO data through scalable and customisable environments. This use case supports the execution of complex data processing workflows, enabling users to preprocess, transform, and analyse EO data effectively.

Key Components

- User Workspace: providing secure environments for executing data processing tasks on existing cloud-based processing platforms.
- Interactive Development Environment (IDE): Supporting data processing scripts and workflows, including pre-configured tools for EO tasks.
- Project Portal: Integration of visualisations into project portals to showcase data insights.
- Product Catalogue: Streamlined data ingestion and access to a wide range of EO datasets.
- Geospatial Explorer and Web Apps: Interactively visualise results.

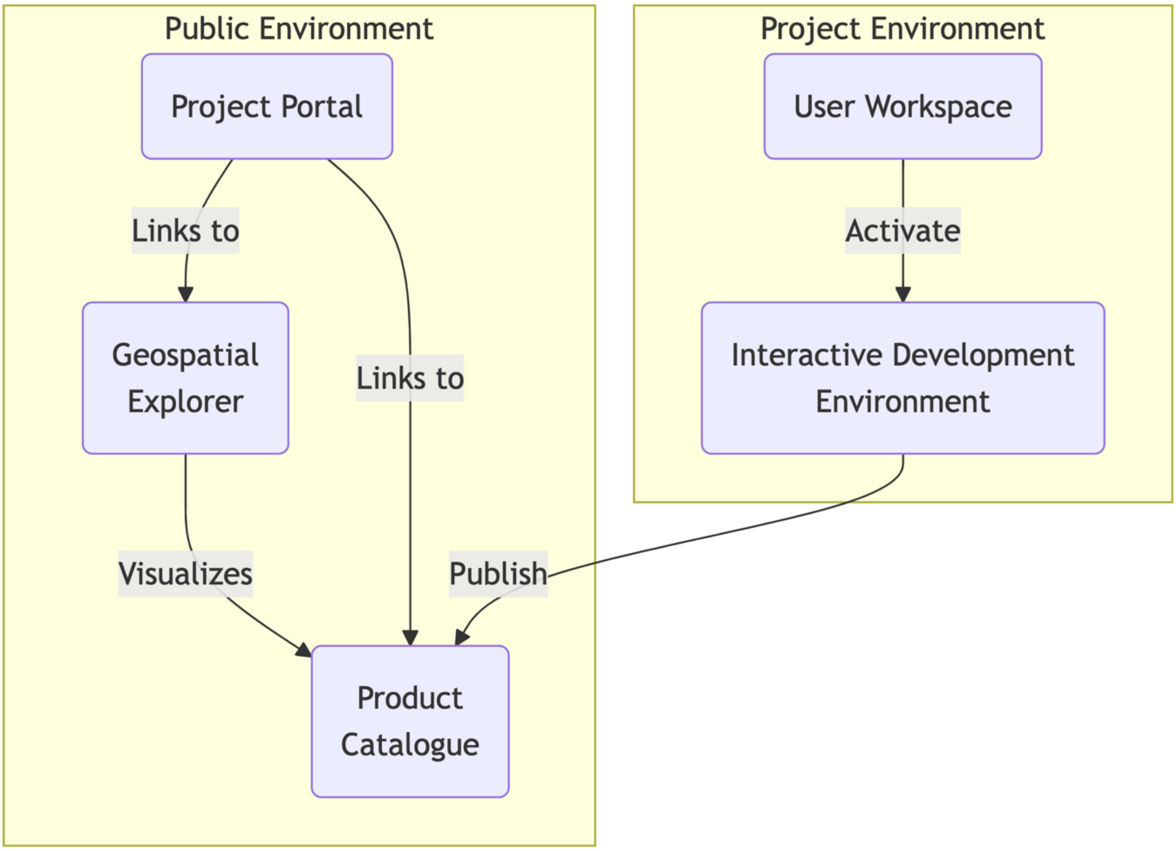


Figure 7. APEx Project Tools - Data Processing

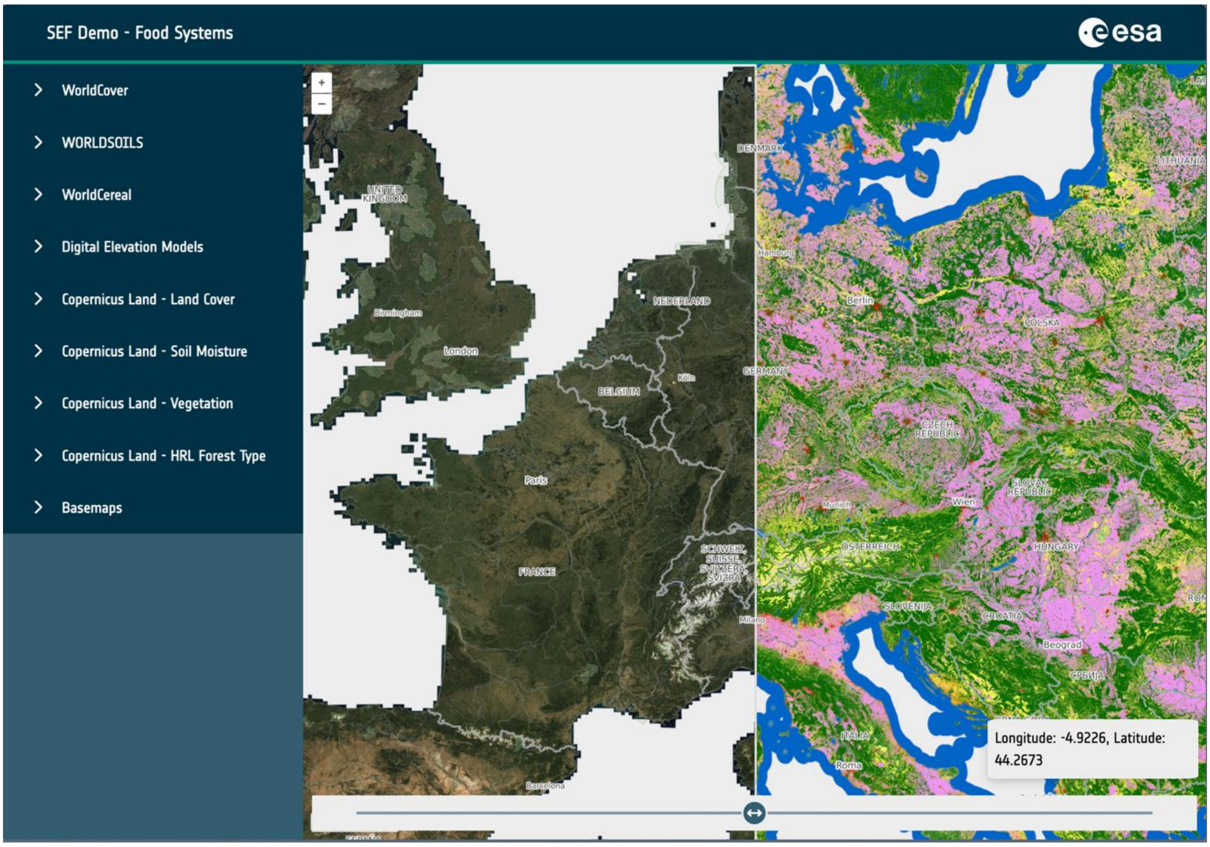
Expected Benefits

- Improved efficiency in data processing workflows.
- Scalable solutions for handling large volumes of EO data.
- Enhanced ability to preprocess and prepare data for analysis and visualisation.

3 GEOSPATIAL EXPLORER

3.1 Overview

The Geospatial Explorer service provides an interactive web front end that can be used for the display and visualisation of geospatial and tabular data ingested from web services following common interoperable protocols (e.g. OGC Standards, STAC, etc.). The service will be data-driven, allowing administrators to define the configuration of the explorer in JSON (i.e. the data layers and functional operations possible for each layer). This configuration will determine how the user interface is rendered at run time and the resulting data and functionality that is exposed to the end user.



Longitude: -4.9226, Latitude: 44.2673

Figure 8. APEx Geospatial Explorer

Typical functions will include the ability to visualise EO data, derived products and associated vector layers (e.g. administrative boundaries), with control over layer ordering, transparency, product comparisons (split screen) and support for features such as cursor inspection, queries, distance measurements and visualisation of tabular data via integrated charts and graphs. The UI will also provide access to metadata records associated with the data that is rendered in the explorer service.

3.2 Showcase Scenarios

The showcase scenarios for the Geospatial Explorer will target both project and SEF use cases.

- Project teams will be able to configure their own explorer interface with data outputs from project work to support the dissemination of project outcomes to project team members, ESA stakeholders and wider interest groups. The explorer will be a key tool for disseminating project deliverables throughout the lifecycle of the project.

- The SEF team will be able to configure the Geospatial Explorer to support the wider promotion of ESA work to policymakers and end users, with the expectation that it will be used at events such as trade shows, conferences and workshops. It is expected that the explorer will be configured around specific “narratives” within overarching SEF “themes” such that the SEF team have a collection of explorer configurations that can be taken to different events for demonstration.

Based on the analysis of various existing ESA and independent third-party tools, it is proposed that there are three levels of Geospatial Explorer capability, as outlined in the following sections. It is important to note that these levels should not be seen as separate entities; functionalities can be mixed and matched within a single explorer instance.

Data Visualization/Exploration (Low Curation)

Features:

- Map with Sidebar layout
- Multiple data layers
- Selectable/toggleable "layer cards"
- Limited layer information on activation

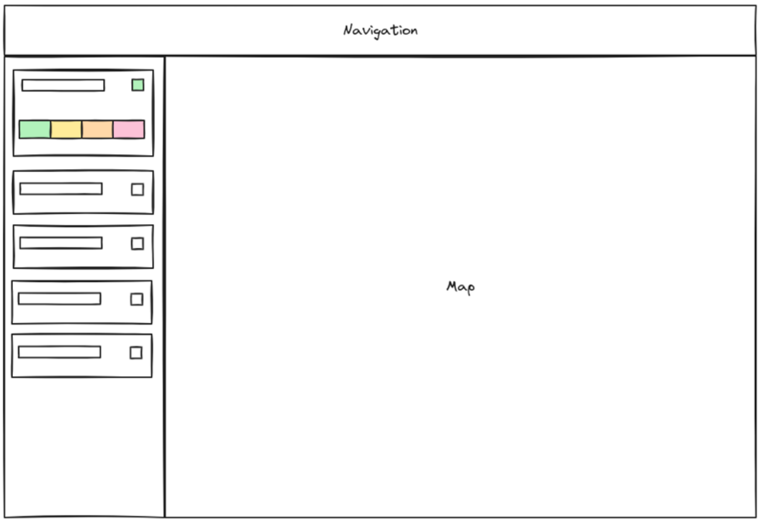


Figure 9. Mock-up of data visualisation/exploration dashboard

Data Analysis (Medium Curation)

Features:

- Full-screen map with popout OR map & sidebar layout
- Focussed set of data/layers
- Selectable/Toggleable (Optionally)
- Sidebar/Popouts with extra layer details, such as:
- Charts
- Data/Time Filtering

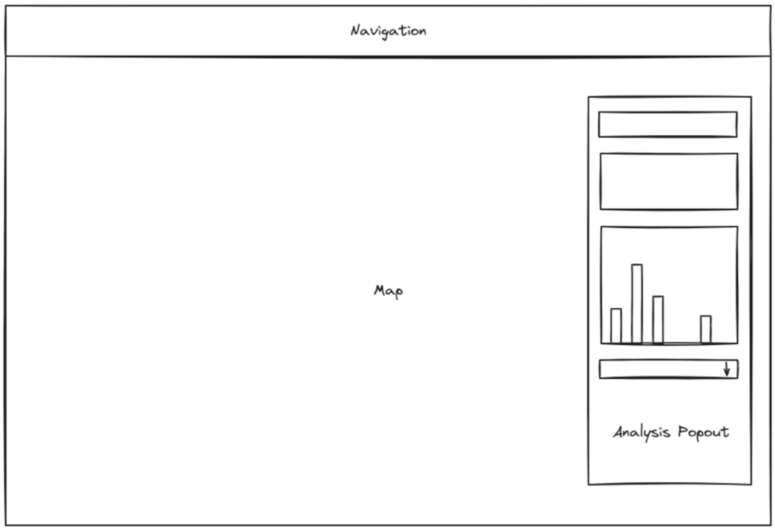


Figure 10. Mock-up of data analysis dashboard

Timeline (High Curation)

Features:

- Map with timeline controls
- Constrained to specific data/areas
- Popout/Sidebar with static content

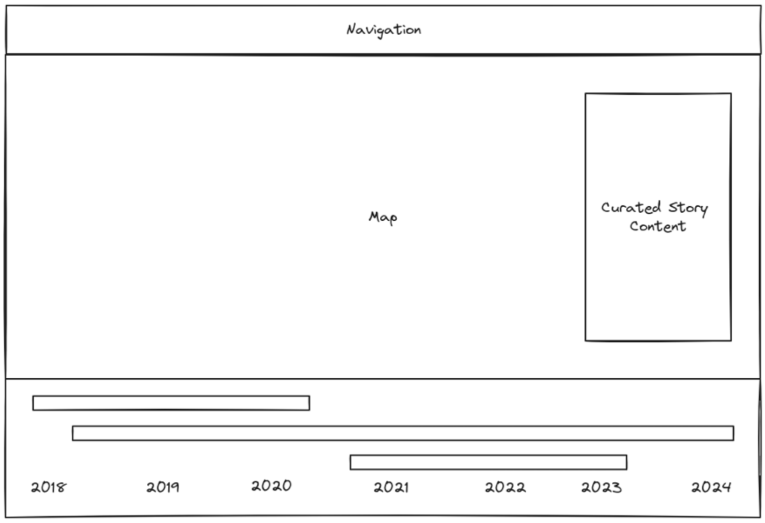


Figure 11. Mock-up of timeline dashboard

3.3 Technical Architecture

The infrastructure requirements for the Geospatial Explorer are ultimately quite simple, as the majority of this APEx component exists on the client side using the React framework. Infrastructure needs for the Geospatial Explorer are:

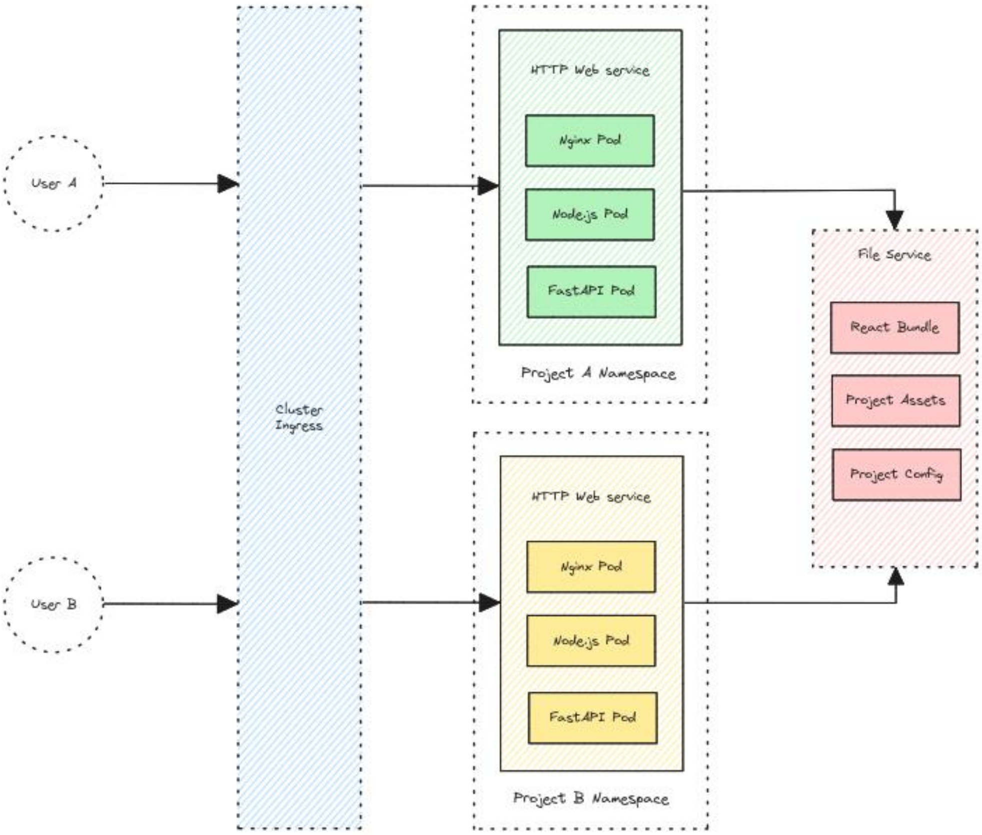


Figure 12. Technical architecture of the Geospatial Explorer Service

- **Cluster Ingress:** External traffic will need to be routed to the explorer web service. This will either need to be based on a domain, subdomain or path basis.

- **Namespace**: Projects could have a unique namespace for isolation purposes.
- Web Service: One or more pods runs a simple nginx web server to provide users with requested resources from the file service.
- **File Service**: Storage for assets and configuration, these will be both generic and project-based. Generic files are for use in every instantiated project and would include the React bundle and any generic HTML documents, whereas the project-based files would include the explorer configuration and project assets for use in branding, for example.

It is important to highlight that the complexity of the required web server and the technology it uses is dependent on further requirement gathering and architectural decisions.

3.4 Operational Management

The Geospatial Explorer service is relatively low maintenance regarding infrastructure and monitoring once the explorer is instantiated successfully.

The scaling of each explorer instance is dependent on traffic levels to the web service hosting it. As the explorer uses a Kubernetes-based infrastructure, the scaling is managed by the cluster based on available resources for the pod.

The use of an analytics platform/framework would be useful for monitoring front-end health however, this task has not been planned. The only concern for monitoring on the front end would be to verify that all calls to third-party data sources are correctly responding. Maintaining these data sources is, however, out of the scope of responsibility for the explorer service.

3.4.1 Configurations

The Geospatial Explorer is configured through a JSON object that is retrieved upon the initial loading of the client-side application. This configuration is hosted by the same service that provides the assets for the explorer and is exposed through a relative `/config` path at the instantiated explorer’s URL. This JSON object is either:

- Created when deployed - Extracted from the configuration of the Kubernetes config map. A default configuration is set in each instance if the config map is not changed.
- Retrieved from an external source by configuring a URL to a valid JSON resource when deploying an instance.

Descriptions and an explanation of the configuration schema can be found in the interoperability guidelines on the APEx Documentation Portal.

Examples of configuration for the explorer can be found in the APEx Geospatial Explorer Config repository on GitHub.

3.5 Usage

This Geospatial Explorer service has been instantiated for the following projects:

Table 1. Usage of Geospatial Explorer

Project	Description	Status	URL
Stakeholder Engagement Facility (SEF)	To promote project results, the SEF has requested the instantiation of a Geospatial Explorer tailored to its key themes. The current focus is on setting up a Geospatial Explorer specifically for the Food Systems and Ecosystems, and Biodiversity themes.	Done	https://explorer.sef-food.apex.esa.int/ https://explorer.sef-ecosystems.apex.esa.int/
APEx	APEx has its own dedicated instance of the Geospatial Explorer that is used to showcase its capabilities and explore the onboarded datasets. The configuration will be updated with each Geospatial Explorer release to ensure that it showcases all new features as and when available.	Done	https://explorer.apex.esa.int/
APEx Demo	To showcase the APEx capabilities, a demo environment has been instantiated, including a Geospatial Explorer.	Done	https://explorer.demo.apex.esa.int/

3.6 Roadmap

The roadmap for the Geospatial Explorer is actively maintained on GitHub at https://github.com/orgs/ESA-APEx/projects/3/views/4 and will be updated with additional features and progress updates when available.

Version	Release Date	Key Features	Notable Changes	Status
v0.0.2	2025-01-10	Initial release, Basic user interface with support for the following sources: • XYZ APIs • GeoJson • WMS/WMTS • COG • WFS • STAC	First stable version deployed	Released

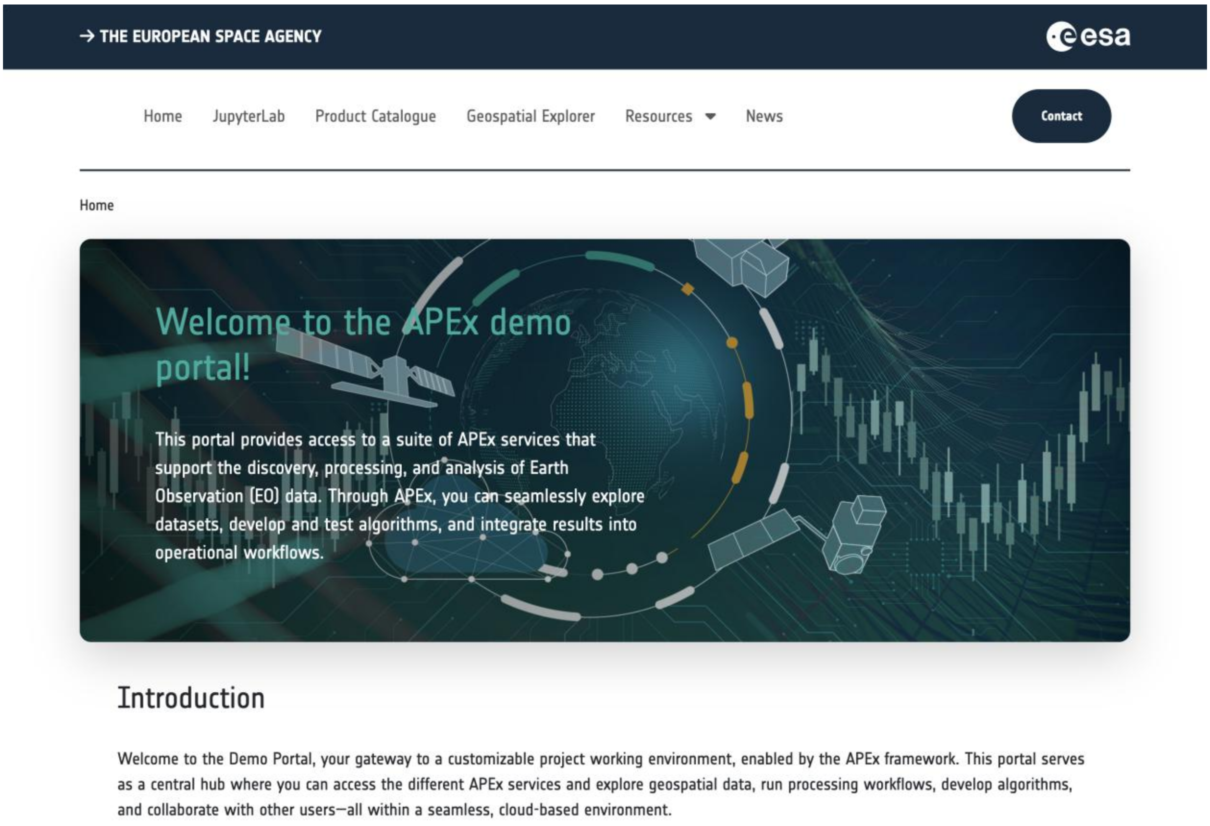
v0.1.0	2025-01-13	Deployment workflows and default configuration	Numerous bug fixes and refactoring	Released
v1.0.1	2025-03-05	Opacity slider controls, Mouse cursors coordinates and scale line, support for external configuration, COG value querying through TITiler, Initial map configuration, source metadata, base layer configuration	Large breaking change with numerous feature releases and several fixes and refactoring.	Released
v1.0.2	2025-03-06	Bug fixes	Addressed an issue with legends not rendering	Released
v2.0.0	TBD	Support for multiple data sources and layer groups, Support for statistical layers and statistics visualisation UI		Development
v2.1.0*	TBD	Support for time series sources and UI for interacting with time-based layers		Planned
v2.2.0*	TBD	Time series statistical layers and statistics visualisation		Planned
v2.3.0*	TBD	Workflow execution		Planning
v2.4.0*	TBD	Raster contrast/colour controls		Planning

* Semantic versioning based on expected change. Version numbers may be different due to changes in the release schedule and the overall impact of changes.

4 PROJECT PORTAL

4.1 Overview

The Project Portal Service provides a solution designed to support the needs of projects by providing a project website upon request. The portal, structured on a predefined base template, offers the flexibility to further customise the overall user experience. Beyond the initial theming options, the service incorporates a Drupal Content Management System (CMS), empowering project members to perform advanced customisations and efficiently manage portal content.



Welcome to the APEx demo portal!

This portal provides access to a suite of APEx services that support the discovery, processing, and analysis of Earth Observation (EO) data. Through APEx, you can seamlessly explore datasets, develop and test algorithms, and integrate results into operational workflows.

Introduction

Welcome to the Demo Portal, your gateway to a customizable project working environment, enabled by the APEx framework. This portal serves as a central hub where you can access the different APEx services and explore geospatial data, run processing workflows, develop algorithms, and collaborate with other users—all within a seamless, cloud-based environment.

Figure 13. APEx Project Portal

The instantiated project portal will have the option to work in synergy with other services offered within APEx. This collaborative approach enables projects to extend their portal by instantiating other APEx services, such as a STAC catalogue, a Geospatial Explorer, a user forum, and more, and linking them to the main project portal, resulting in a dedicated, streamlined project ecosystem.

4.2 Showcase Scenarios

The project portal can bring a lot of value to projects, including:

1. **Efficient project dissemination and community engagement:** Facilitates the rapid and broad dissemination of project information, ensuring that updates, results, and important announcements reach the intended audience quickly and effectively.

2. **Professional branding with customisation possibilities:** The portal comes with polished, consistent branding out of the box, enhancing the project's credibility and professionalism and fostering trust among stakeholders and the public. Additionally, project teams can tailor the portal’s appearance to align with specific branding guidelines or aesthetic preferences, creating a unique and engaging user experience that reflects the project's identity.

3. **Content management and moderation:** The portal's robust content management capabilities enable teams to streamline the creation, moderation, and updating of content. Advanced features like content moderation workflows and version control ensure that all shared information is accurate, current, and meets our high-quality standards. The system supports various roles and permissions, including administrators and content creators, fostering a self-regulated environment. This empowerment allows project teams to manage their content and user access autonomously, thereby enhancing operational efficiency.

4.3 Technical Architecture

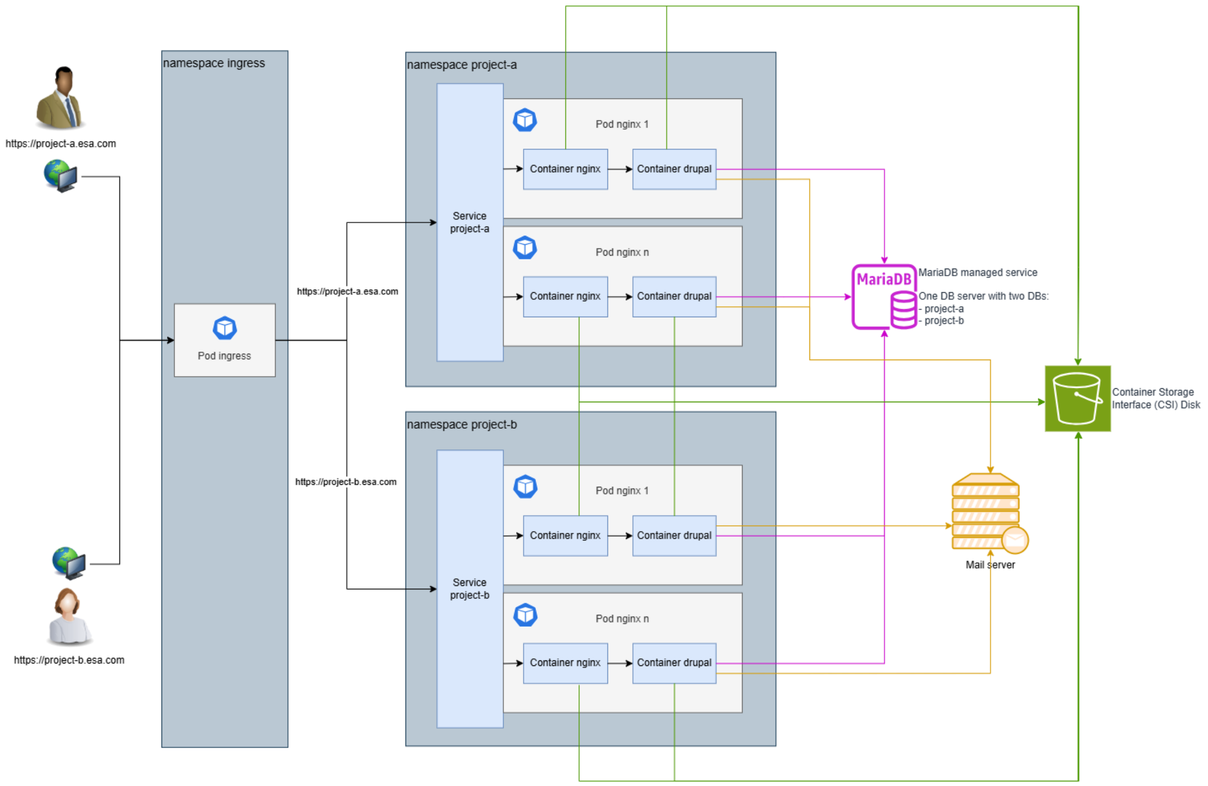


Figure 12. Technical architecture of the Project Portal Service

This infrastructure is designed to host Drupal websites for multiple projects, each within its own Kubernetes namespace. The setup ensures isolation, scalability, and efficient management of resources. The main components include an ingress controller, various services and pods within project namespaces, a MariaDB-managed service, an OpenID Connect Identity Provider, and NFS storage.

The overall architecture is defined by the following components:

1 **Ingress:** This component is responsible for routing external traffic to the appropriate project namespace based on the URL. It uses rules to direct traffic to *project-x.apex.esa.int*.

- **Namespace**: Each project (e.g., *project-a.apex.esa.int* and *project-b.apex.esa.int*) has its own namespace for resource isolation. Each instantiated namespace will spawn one or more pods with the following containers:
    - **nginx**: Each nginx container handles HTTP requests and forwards them to the appropriate Drupal container using FastCGI (FPM).
    - **Drupal:** This container balances the load across multiple Drupal pods within the namespace. Each container contains the Drupal application running with PHP-FPM (FastCGI Process Manager), which processes PHP scripts.

2 **MariaDB Managed Service**: This is a centralised managed database service. Each project namespace has its own database instance within this service.

3 **OpenID Connect Identity Provider**: This component handles user authentication via the OpenID Connect protocol, ensuring secure access to the Drupal websites. It communicates with the Drupal pods over HTTPS.

- **CSI**: Provides persistent storage for the Drupal applications and nginx pods, allowing them to store and share files. This is essential for storing user-uploaded content, configuration files, and other persistent data.
- **Mail server**: The Drupal containers use this server to send notification e-mails.

4.4 Operational Management

The project portal operates within a Kubernetes environment shared across all APEx Instantiation Services. Based on prior experiences, each project portal will receive sufficient resources for hosting a basic website. The initial setup anticipates minimal scaling needs, as the basic configuration will meet the requirements of most projects.

However, throughout the project lifecycle, the team will explore optional enhancements to operational management, including:

Hoofdstuk 1 Implementing redundant Drupal nodes to ensure high availability.
Hoofdstuk 2 Enhancing performance by integrating Redis components for Drupal caching.

Continuous monitoring of the Kubernetes cluster ensures optimal performance and availability of the instantiated project portals. Automated maintenance tasks, such as updates and backups, are conducted to minimise downtime and uphold data integrity. This approach establishes a robust and dependable environment for users.

4.5 Usage

This Project Portal service has been instantiated for the following projects:

Table 2. Usage of Project Portal

Project	Description	Status	URL
APEx	The main APEx website is using an instantiated project portal which is fully customized to host the website content.	Done	https://apex.esa.int/
APEx Demo	To showcase the APEx capabilities, a demo environment has been instantiated, including a project portal.	Done	https://demo.apex.esa.int/
World Ecosystem Extent Dynamics (WEED)	A project portal has been created for the WEED project to replace the existing website.	Done	https://weed.apex.esa.int/en

4.6 Roadmap

The following features have been identified by the team to ensure the ongoing evolution of the project portal service:

Table 3. Project Portal Roadmap

Feature	Description	Status
Enhanced customisation	Improve the ability for project administrators to easily customise the project portal to meet their specific needs.	In Progress
Extended capabilities	Expand the portal's options based on user feedback from instantiated projects. Some of the currently identified features are the creation of an events page and the publication of news items.	In Progress
Enhanced integration with IaaS components	Expand the portal's options based on user feedback from instantiated projects. Some of the currently identified features are the creation of an events page and the publication of news items.	Done

5 USER WORKSPACE

5.1 Overview

The User Workspace service within the APEx Instantiation Services provides secure, personalised environments for individual users to perform a wide range of tasks, including development, data processing, visualisation, and analysis. These single-user environments are managed by JupyterHub and dynamically provisioned using Kubernetes, ensuring scalability, isolation, and ease of use.

It is designed to provide the following key benefits to users when accessing their instance:

- **Flexibility and customisation**
  Configure a workspace with tools and settings tailored to specific needs.
- **Optimised for EO tasks**
  Access specialised tools and libraries for Earth observation.
- **Integrated ecosystem**
  Utilise other APEx services through your workspace.
- **Collaborative work**
  Share workspaces within team to facilitate collaboration and data sharing. Enhance community engagement and knowledge sharing by publishing results.

Key features and capabilities of the User Workspace include:

- **Cloud services orchestration**: High availability and performance of multi-user-facing services, via automated deployment and scaling of containerised applications, as well as built-in load balancing and service discovery mechanisms.

- **Secure, Isolated Environments**: Each user is provided with a dedicated namespace within the Kubernetes cluster, ensuring resource isolation and secure data management.

- **Customisable Workspaces**: Workspaces can be tailored to user needs with specific configurations and tools, including pre-configured environments for tasks like data analysis and machine learning.

- **Seamless APEx integration**: Workspaces benefit from a generic, configurable framework, ensuring the delivery of a broad range of APEx applications and services, empowering efficient user workflows.

- **EO-focused tools access**: Readiness for resource-intensive software libraries like SNAP and GDAL, which are specifically designed for Earth observation tasks.

- **Robust data management**: Secure, persistent storage and retrieval of data using PersistentVolumeClaims (PVCs), along with data management tools, connectivity to external data sources, and data sharing mechanisms.

5.2 Showcase Scenarios

The User Workspaces support a variety of use cases, making them versatile tools for the EO community. Some typical scenarios include:

- **Development and Testing**: Researchers and developers can use the User Workspaces to develop and test new algorithms and models. For instance, a user might leverage the IDE integrated with JupyterLab to write and debug Python scripts for processing satellite imagery.

- **Data Analysis and Visualisation**: Analysts can perform exploratory data analysis and create visualisations using tools like JupyterLab or QGIS. For example, an analyst might use JupyterLab to analyse climate data and visualise trends over time.

- **Educational and Training Purposes**: The User Workspaces can be used to create interactive tutorials and practical assignments that guide users through various aspects of APEx, from setting up workspaces and accessing data in the Product Catalogue to using the IDE for development tasks.

5.3 Technical Architecture

The User Workspaces are built on advanced technologies and frameworks to ensure robustness and scalability. The Application Hub, extending on JupyterHub, provides a publicly accessible web interface for authenticated and authorised users to instantiate their POD instances securely. KubeSpawner, a Kubernetes-native spawner for JupyterHub, dynamically creates and manages servers within the Kubernetes cluster.

When a user launches their server, several Kubernetes resources are created. Each user is allocated a dedicated namespace, providing a secure and organised virtual environment within the Kubernetes cluster. Configuration data is stored separately from the application's code in a ConfigMap, which can include Conda environment configurations. PersistentVolumeClaims (PVCs) are created to request storage resources, specifying the necessary size and access modes for mounting volumes with read/write access. Finally, a POD is created and started using a dedicated application profile based on the user's selection.

A PostgreSQL database is used to store user information for JupyterHub, managing authentication and user data. Data management involves secure storage and retrieval facilitated through PVCs and integration with external data sources. Users can access and share data within the workspace efficiently.

Workspace setup and customisation allow users to configure their environments with specific tools and configurations required for their tasks. Integration with other APEx services, such as the IDE for enhanced development capabilities or specialised Dashboards, further enhances the functionality of the User Workspaces. Security and compliance measures ensure that only authenticated users can access their workspaces, with data privacy and regulatory compliance measures in place to protect user data.

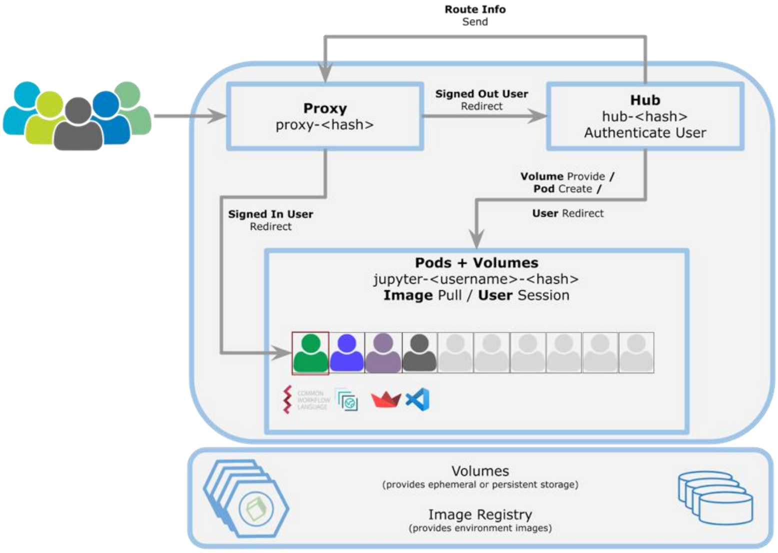


Figure 13. Workflow and Architecture of User Workspaces

The diagram above illustrates how user sessions are created, authenticated, and managed within the Kubernetes infrastructure. It highlights the roles of the Proxy, Hub, PODs, volumes, and Image Registry in providing secure, isolated, and customised environments for individual users. The system ensures efficient resource management and a seamless user experience for various tasks, including development and data analysis.

Kubernetes plays a crucial role in managing the deployment, scaling, and operation of containerised applications such as the APEx-selected IDE or GIS solutions. By ensuring auto-scalability, self-healing, and load distribution, Kubernetes provides a stable and efficient environment for development tasks. JupyterHub orchestrates the creation and management of user-specific development environments, allowing seamless integration with the broader APEx services and enabling the IDE to function smoothly within the APEx ecosystem.

Data management is a key aspect of the IDE and GIS software solutions. They enable the integration with various data sources, including external databases and APEx's Product Catalogue, enabling users to access and utilise these data sources within their development workflows. Data generated or used within the IDE or GIS User Workspace is securely stored using PersistentVolumeClaims (PVCs) and can be retrieved as needed, ensuring efficient and secure data management.

Application portability is a cornerstone of the proposed approach, achieved through containerisation technologies. Developers can encapsulate applications with all essential configurations and dependencies, ensuring consistent behaviour across diverse deployment environments.

Security and compliance are prioritise. User authentication and authorisation are managed through JupyterHub, ensuring that only authorised users can access the solution. User data is stored in isolated environments, adhering to data privacy regulations and standards. Configurations and environment settings are managed securely to prevent unauthorised access, ensuring that the development environment remains secure and compliant.

5.4 Operational Management

Deployment and scaling of User Workspaces are managed by Kubernetes, which handles the scaling of resources, allowing the platform to accommodate varying workloads and user demands efficiently. Continuous monitoring of the Kubernetes cluster and JupyterHub instances ensures optimal performance tuning and service availability. Automated maintenance tasks, such as updates and backups, minimise downtime and ensure data integrity, providing a robust and reliable environment for users.

5.5 Roadmap

The User Workspace service has been available since Q4 2024. It was started as an MVP (Minimum Viable Product), already supporting the range of user activities for collaboration (i.e. “collective progress within the Earth Observation R&D community”) and project results sharing. It currently supports the instantiation of User Workspaces configured for QGIS, Code Server IDE or JupyterLab IDE.

The following features have been identified for the evolution of the service:

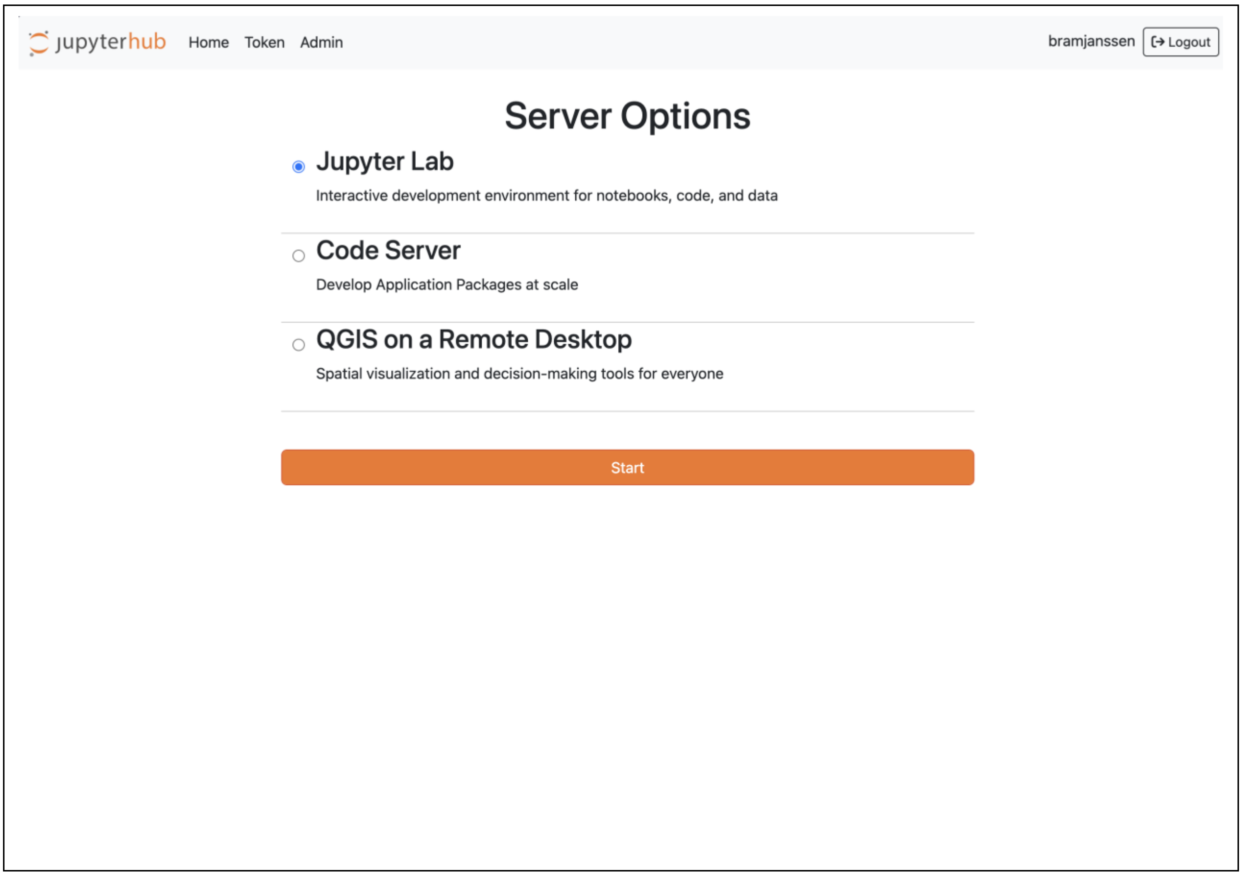
Table 4. User Workspace Roadmap

Feature	Description	Status
Enhanced stability	Fine tuning of the Kubernetes Cluster Autoscaler for performances and stability of cluster instances. Schedule: Q1 2025	Done

6 IDE & GIS USER WORKSPACES

6.1 Overview

The Interactive Development Environment (IDE) capacity within the APEx Instantiation Services primarily leverages the power of the Code Server software (Visual Studio Code in the Cloud), and of the JupyterLab software. The Geographic Information System (GIS) capacity leverages QGIS as a Remote Desktop software.



Server Options

● Jupyter Lab
Interactive development environment for notebooks, code, and data

○ Code Server
Develop Application Packages at scale

○ QGIS on a Remote Desktop
Spatial visualization and decision-making tools for everyone

Figure 1414. Current APEx IDE and GIS Services

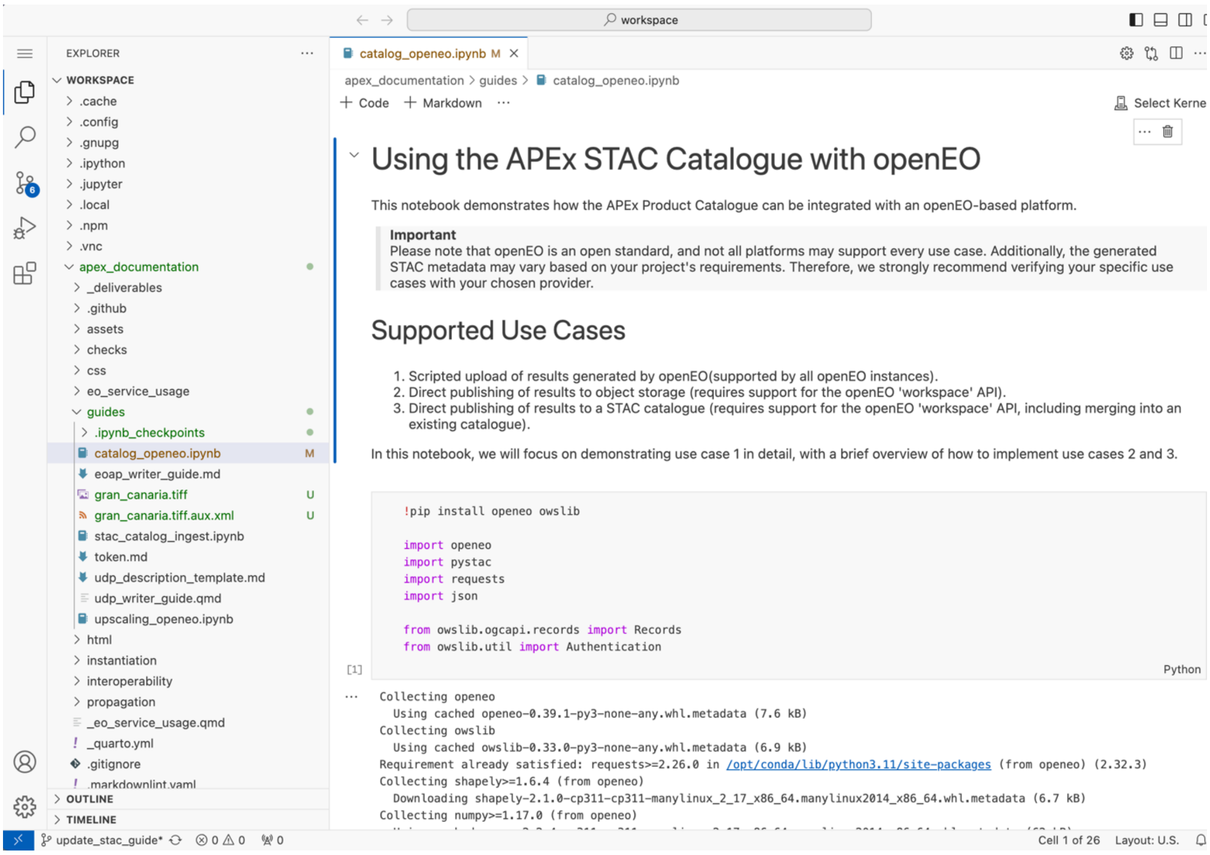
These APEx User Workspace services allow users to maintain a familiar environment and rich feature set while benefiting from the power and resources of server-side computing. This is particularly advantageous for those working on resource-intensive tasks or needing access to a consistent development environment from various locations and devices.

The server-based nature ensures that developers are not constrained by their local machine's hardware capabilities, allowing them to harness the computational power of remote servers.

Tailored specifically for EO tasks, this environment furnishes developers with an array of tools and libraries fine-tuned for programming languages and productivity plugins or extensions.

6.1.1 Code Server

The APEx Code Server solution is an Integrated Development Environment delivered as a cloud-based user workspace, tailored to support the activities of Earth observation (EO) projects.



Using the APEx STAC Catalogue with openEO

This notebook demonstrates how the APEx Product Catalogue can be integrated with an openEO-based platform.

> **Important**
> Please note that openEO is an open standard, and not all platforms may support every use case. Additionally, the generated STAC metadata may vary based on your project's requirements. Therefore, we strongly recommend verifying your specific use cases with your chosen provider.

Supported Use Cases

1. Scripted upload of results generated by openEO(supported by all openEO instances).
2. Direct publishing of results to object storage (requires support for the openEO 'workspace' API).
3. Direct publishing of results to a STAC catalogue (requires support for the openEO 'workspace' API, including merging into an existing catalogue).

In this notebook, we will focus on demonstrating use case 1 in detail, with a brief overview of how to implement use cases 2 and 3.

```
!pip install openeo owslib

import openeo
import pystac
import requests
import json

from owslib.ogcapi.records import Records
from owslib.util import Authentication
```

[1] Python

```
Collecting openeo
  Using cached openeo-0.39.1-py3-none-any.whl.metadata (7.6 kB)
Collecting owslib
  Using cached owslib-0.33.0-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: requests>=2.26.0 in /opt/conda/lib/python3.11/site-packages (from openeo) (2.32.3)
Collecting shapely>=1.6.4 (from openeo)
  Downloading shapely-2.1.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
Collecting numpy>=1.17.0 (from openeo)
```

Figure 1515. APEx IDE - Code Server

The Code Server IDE within the APEx Instantiation Services is built on the User Workspace architecture, leveraging Kubernetes and JupyterHub for orchestration and management.

Each Code Server workspace comes equipped with the Visual Studio Code Server, an extension of Microsoft's popular VS Code editor, as well as with a private data products catalogue. These features empower developer users to edit and build EO data processing algorithms and workflows, accelerating project outcomes within a dedicated, tool-rich environment.

The Code Server setup encapsulates all the capabilities of Microsoft's popular VS Code editor and extends them to be run and accessed on a remote server. Beyond the core functionality of its desktop counterpart, the Code Server IDE offers additional features tailored for remote development, such as integrated Git support, debugging tools, and a plethora of extensions (for Code Server, from the VS Code Marketplace and for JupyterLab, from the PyPI.org registry). It provides support for programming languages like Python, R, and Java. Key libraries such as SNAP and GDAL are integrated, providing robust capabilities for EO data discovery, access, processing, and analytical needs.

It seamlessly adapts to containerised environments, enabling developers to create, test, and deploy applications within isolated, replicable, and consistent environments, ensuring consistent behaviour across development, staging, and production phases.

Code Server forms the core development environment, delivering a comprehensive code editing and debugging interface. This server-based IDE allows developers to maintain the familiar environment and feature set of VS Code while benefiting from the power and resources of server-side computing. The server-based nature ensures that developers are not constrained by their local machine's hardware capabilities, enabling them to leverage the computational power of remote servers

Code Server serves as the core development environment, providing a powerful and flexible platform for coding and debugging, allowing developers to harness server-side computational power while maintaining a familiar interface.

Key Features

The APEx Code Server workspace includes several features that make it suitable for both individual researchers and teams working on EO research and applications.

- **Customisable workspace**
  Configure your workspace with tools and settings tailored to your specific needs, including debugging tools and machine learning, as well as a plethora of extensions for Code Server from the VS Code Marketplace.

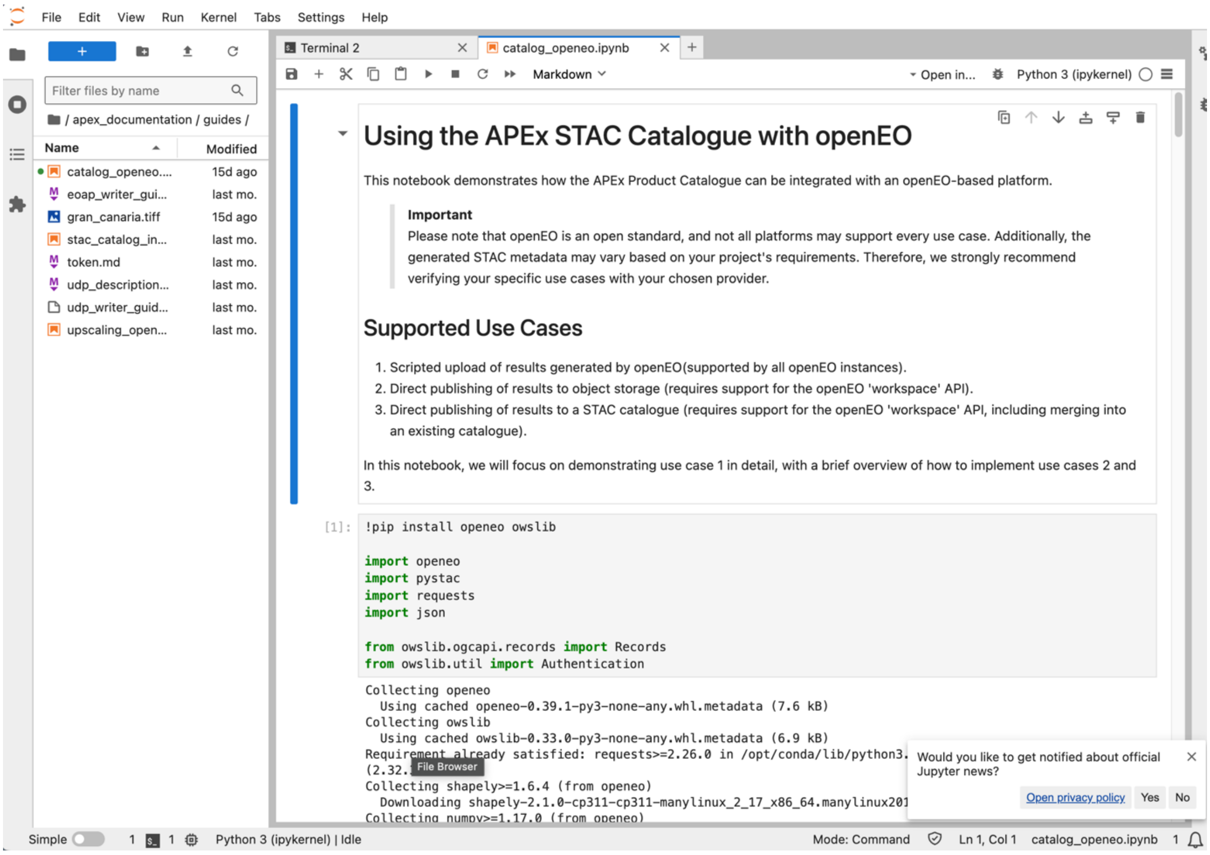
- **EO-focused tools**
  Access integrated libraries like SNAP and GDAL, which are specifically designed for Earth observation tasks.

- **Support for every major programming language**
  Native support for a wide range of programming languages, including JavaScript, TypeScript, CSS, HTML, C++, C#, and Python, complemented by a vast repository of extensions for virtually any language or file format (Java, Go, PHP, Ruby, ...)

- **Remote development and collaborative work**
  Leverage integrated Git support, allowing team members to work on the same project by tracking changes and merging updates, helping to keep the code organised while avoiding conflicting code changes.

6.1.2 JupyterLab

The JupyterLab setup encapsulates a web-based interactive development environment for Jupyter notebooks, code, and data. It is the user interface for *Project Jupyter*, offering a flexible user interface and more features than the classic notebook UI. It is a web application providing a development environment in which processing algorithms and services can be developed, tested, and debugged. JupyterLab supports execution environments (called "kernels") in several dozen languages, including Julia, R, Haskell, Ruby, and Python (via the IPython kernel). It seamlessly adapts to containerised environments, enabling developers to create, test, and deploy applications within isolated, replicable, and consistent environments, ensuring consistent behaviour across development, staging, and production phases.



Name	Modified
catalog_openeo....	15d ago
eoap_writer_gui...	last mo.
gran_canaria.tiff	15d ago
stac_catalog_in...	last mo.
token.md	last mo.
udp_description...	last mo.
udp_writer_guid...	last mo.
upscaling_open...	last mo.

Using the APEx STAC Catalogue with openEO

This notebook demonstrates how the APEx Product Catalogue can be integrated with an openEO-based platform.

> **Important**
> Please note that openEO is an open standard, and not all platforms may support every use case. Additionally, the generated STAC metadata may vary based on your project's requirements. Therefore, we strongly recommend verifying your specific use cases with your chosen provider.

Supported Use Cases

1. Scripted upload of results generated by openEO(supported by all openEO instances).
2. Direct publishing of results to object storage (requires support for the openEO 'workspace' API).
3. Direct publishing of results to a STAC catalogue (requires support for the openEO 'workspace' API, including merging into an existing catalogue).

In this notebook, we will focus on demonstrating use case 1 in detail, with a brief overview of how to implement use cases 2 and 3.

```
[1]: !pip install openeo owslib

import openeo
import pystac
import requests
import json

from owslib.ogcapi.records import Records
from owslib.util import Authentication
```

```
Collecting openeo
  Using cached openeo-0.39.1-py3-none-any.whl.metadata (7.6 kB)
Collecting owslib
  Using cached owslib-0.33.0-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: requests>=2.26.0 in /opt/conda/lib/python3.
(2.32.
Collecting shapely>=1.6.4 (from openeo)
  Downloading shapely-2.1.0-cp311-cp311-manylinux_2_17_x86_64.manylinux201
Collecting numpy>=1.17.0 (from openeo)
```

Would you like to get notified about official Jupyter news?

Open privacy policy Yes No

Figure 1616. APEx IDE - JupyterLab

Key Features

The APEx JupyterLab workspace includes several features that make it suitable for both individual researchers and teams working on EO research and applications.

- **Customisable workspaces**
  Configure your workspace with tools and settings tailored to your specific needs, including debugging tools, data analysis, and data visualisation, as well as a plethora of extensions for JupyterLab from the PyPI.org registry.

- **EO-focused tools**
  Access integrated libraries like SNAP and GDAL, which are specifically designed for Earth observation tasks.

- **Support for every major programming language**
  Native support for over 40 preinstalled execution environments (called "kernels") including Python, R, Julia, Scala, Haskell and Ruby, complemented by a vast

repository of additional kernels maintained by the open source software community with various levels of support.

- **Remote development and collaborative work**
  Leverage integrated Git support, allowing team members to work on the same project by tracking changes and merging updates, helping to keep the code organized while avoiding conflicting code changes.

6.1.3 QGIS

The APEx QGIS workspace is a fully-fledged Geographic Information System delivered as a cloud-based user workspace, tailored to support the activities of Earth observation (EO) projects.

Each QGIS workspace comes equipped with the free and open source software project hosted on QGIS.org association, as well as with a private data products catalogue. These features empower developer users to edit and build EO data processing algorithms and workflows, accelerating project outcomes within a dedicated, tool-rich environment.

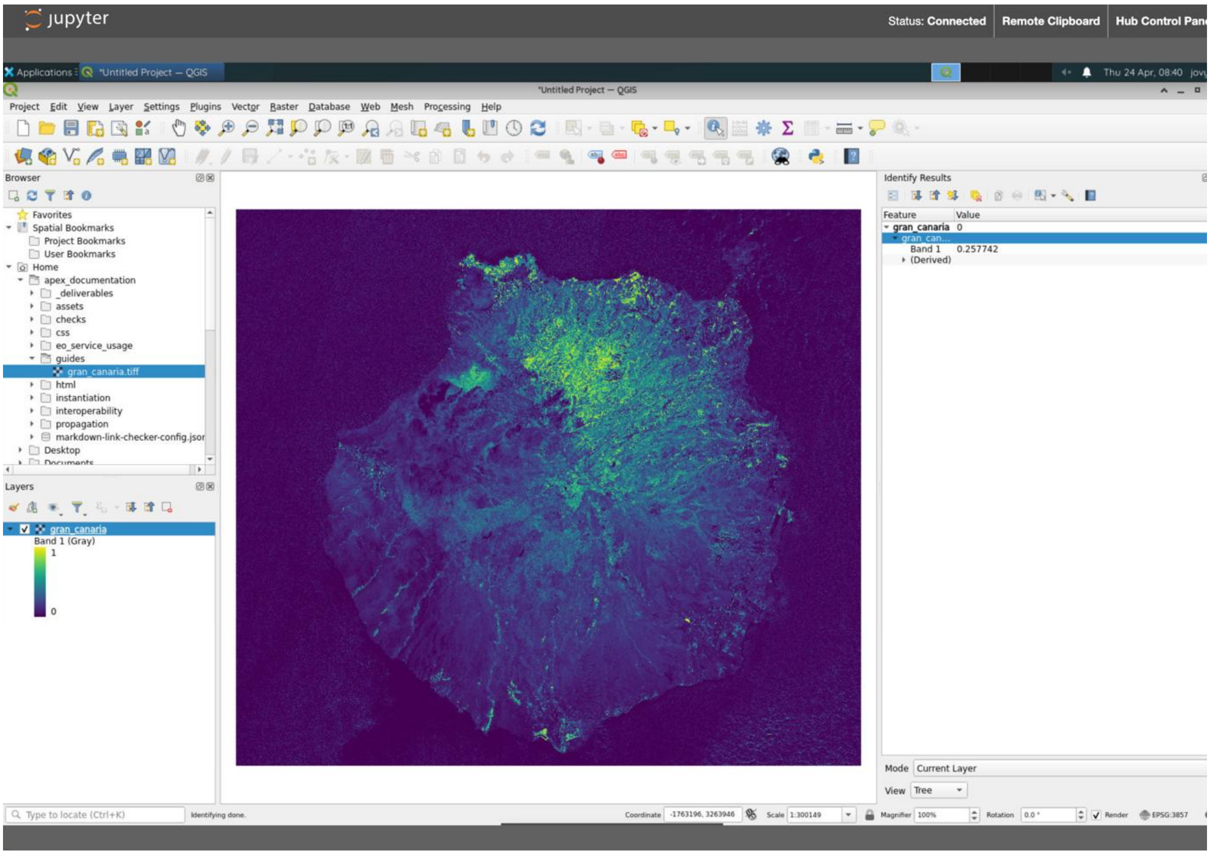


Figure 1717. APEx IDE - QGIS

The APEx QGIS workspace includes several features that make it suitable for both individual researchers and teams working on EO research and applications.

- **Customisable workspaces**

Configure your workspace with tools and settings tailored to your specific needs, including geographic features digitising tools, geospatial analysis workflows and map production options, as well as advanced functions from a plethora of 2000+ plugins developed by the QGIS community.

- **EO-focused tools**
  Access specialised plugins such as for raster data analysis, or for visualisation of multi-sensor time series, all specifically designed for Earth observation tasks.

- **Support for every major GIS toolbox**
  Native support for a wide range of geospatial data management toolboxes, including GRASS and GDAL.

- **Collaborative work**
  Store, manage, and share QGIS projects with the ability to connect to external data sources as needed.

6.2 Showcase Scenarios

The Interactive Development Environment supports a variety of use cases, making it an essential tool for developers, researchers, and data scientists within the EO community. Some typical scenarios include:

- **Algorithm Development and Testing**: Researchers and developers can write, test, and debug new algorithms for processing satellite imagery or other EO data. For instance, a user might develop a script to detect deforestation using multi-temporal satellite images. To support the productivity of developers, both the Code Server solution and the JupyterLab solution use a mechanism of extensions. Code Server extensions are available from the Open VSX Registry.

  Note: there is no GitHub Copilot extension available yet for Code Server or JupyterLab. Nevertheless, there is an equivalent functionality provided by the Code Server extension "Continue - Codestral, Claude, and more", an open-source AI code assistant: https://open-vsx.org/extension/Continue/continue

  It allows to connect any models and any context to build custom autocomplete and chat experiences inside Code Server:

  - "Chat" makes it easy to ask for help from an LLM without needing to leave the Code Server user interface
  - "Autocomplete" provides inline code suggestions as you type
  - "Edit" is a convenient way to modify code without leaving your current file
  - "Actions" are shortcuts for common use cases.

This extension asks for API keys to use the models. This has been successfully tested and could be an option for the APEx use cases.

- **Collaborative Projects**: Teams can work collaboratively on projects, sharing code and resources in real-time. A group of data scientists might collaboratively develop a machine-learning model to predict crop yields based on various data inputs.

- **Data Science Notebooks**: the term "Notebook" usually covers two different concepts, either the user-facing application to edit code and text (this originates from *Project Jupyter*'s software product initially branded "Jupyter Notebooks", nowadays "JupyterLab"), or more commonly the underlying file format which is interoperable across many IDE software solutions. Both solutions proposed for APEx, Code Server, and JupyterLab have multi-kernel language support (Python, R, Ruby, ...). Code Server supports local development of Jupyter Notebooks: the Jupyter extension for VS Code is a very popular extension in the VS Code Marketplace. JupyterLab is a very popular tool within the Open Science community for working with notebooks, with its native support for data science, data visualisation and reproducible environments.

- **Data Processing Pipelines**: Users can develop and test data processing pipelines that automate the ingestion, processing, and analysis of large EO datasets. An example use case could be to create a workflow setting up a pipeline to preprocess satellite images and extract relevant features for further analysis.

Note: The IDE solutions provide the tooling for application developers to implement the Best Practice for Earth Observation Application Package as defined by the Open Geospatial Consortium (OGC 20-089) and the EO Exploitation Platform Common Architecture (EOEPCA) spearheaded by the European Space Agency (ESA). This best practice supports developers in adapting and packaging their existing algorithms to be reproducible, deployed, and executable on different platforms.

An EO Application within the IDE is treated as a command-line interface (CLI) tool that runs as a non-interactive executable program. It receives input arguments, performs a computation, and terminates after producing some output. These applications, written in various programming languages such as Python, Java, C++, C#, and shell scripts, use specific software libraries like SNAP, GDAL, and Orfeo Toolbox. Developers build container images that encapsulate their applications and command-line tools, along with necessary runtime environments, and publish these images on container registries for easy access and deployment.

The IDE supports the use of the Common Workflow Language (CWL), allowing developers to delineate and disseminate application workflows in a recognised format. CWL documents comprehensively describe the data processing application, including parameters, software items, executables, dependencies, and metadata. This standardisation enhances collaboration, clarity, and operational consistency, ensuring that applications are reproducible and portable across various execution scenarios, including local computers, cloud resources, high-performance computing (HPC) environments, Kubernetes clusters, and services deployed through an OGC API - Processes interface.

Version control and continuous integration are integral components of the IDE's technical architecture. The IDE enables access to VCS (e.g. GitLab, GitHub) for efficient code repository management, version control, collaboration, and monitoring of code changes. Automated continuous integration (CI) tools manage the build, test, and deployment tasks in response to code modifications, ensuring that applications are always in a deployable state. This automation minimises manual testing overhead and accelerates the rollout of new features or updates.

6.3 Technical Architecture

The IDE and GIS services within the APEx Instantiation Services are built on the User Workspace service architecture, as described in the previous section.

6.4 Operational Management

The deployment and scaling of the IDE and GIS services within the APEx Instantiation Services are powered by the User Workspace service, as described in the previous section.

6.5 Usage

This IDE service has been instantiated for the following projects*:

Table 5. Usage of Project Portal

Project	Description	Status	URL
Stakeholder Engagement Facility	To support trainings and to explore the capabilities of APEx, the SEF has requested access to the IDEs.	Done	https://apphub.sef-food.apex.esa.int/ https://apphub.sef-ecosystems.apex.esa.int/
APEx Demo	To showcase the APEx capabilities, a demo environment has been instantiated, including the IDEs.	Done	https://apphub.demo.apex.esa.int

(*) Access to these environments will trigger the creation of cloud resources; therefore, they are not publicly available and require a valid APEx account along with permissions to access the project resources.

6.6 Roadmap

The roadmap for this service will evolve with future versions of the document, in particular for the use of extensions and plugins, and will be shaped by user feedback collected during its provision to projects.

7 PRODUCT CATALOGUE

7.1 Overview

The catalogue service enables projects to create their own SpatioTemporal Asset Catalogue (STAC), which serves as a comprehensive metadata collection for efficient organisation and discovery of geospatial data assets. This information is presented in a user-friendly JSON format.

Instantiating a project-specific catalogue gives projects the freedom to use it for a range of purposes, from experimental to operational use cases. Assets that are considered project deliverables will normally be added to the centralised project results repository.

STAC structures data around assets, collections, and catalogues, simplifying search and access based on attributes such as time, location, and other metadata. The concept of a STAC catalogue emphasises simplicity, operating as a REST-based web service with a single HTTP endpoint for communication. This API-centric design ensures easy integration into existing applications and tools, including other APEx services.

7.2 Showcase Scenarios

The product catalogue serves several beneficial scenarios for projects:

- **Data Accessibility:** The service offers easy access to a wide range of geospatial data sources, enabling project teams to quickly locate and utilise relevant data without the need to manage large datasets locally.

- **Tool Integration:** By adhering to the STAC standard, the service enhances interoperability among different data sources, software tools and processing platforms such as openEO and OGC Application Packages. This ensures seamless integration of datasets from various providers, supporting comprehensive analysis and decision-making. This interoperability extends to other APEx services such as the Geospatial Explorer, Interactive Development Environments, User Workspaces….

- **Efficient Search and Discovery:** Users can efficiently search and discover datasets based on specific criteria such as time, location, and data type. This capability streamlines research, planning, and operational tasks by providing quick access to relevant information.

- **Enhanced Collaboration:** The service promotes collaboration by providing a centralised platform where project stakeholders can access and share geospatial data and analyses. This collaborative environment fosters innovation and knowledge sharing across disciplines and organisations.

7.3 Technical Architecture

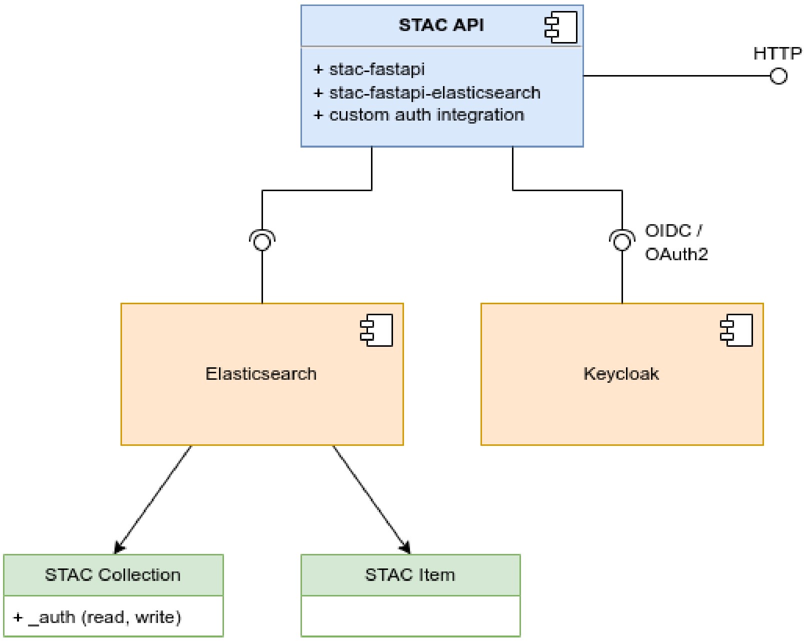


Figure 18. Technical architecture of the Product Catalogue Service

The product catalogue deployment consists of a web service pod and a managed database. The web service, based on STAC *FastAPI*, supports multiple implementations and integrates with various database technologies:

1) https://github.com/stac-utils/stac-fastapi-elasticsearch-opensearch
2) https://github.com/stac-utils/stac-fastapi-pgstac

STAC FastAPI has demonstrated robustness in running production-grade catalogues.

Based on previous experiences, the project team favours the Elasticsearch approach for database implementation. This choice minimises the complexity of database tuning, which typically requires deep knowledge of the database software. Database tuning needs can vary depending on project-specific setups, such as STAC metadata definitions, making a low-maintenance option like Elasticsearch preferable.

It's essential to note that the service may experience high request volumes, particularly when used in conjunction with a viewing service like an APEx Dashboard. Consequently, achieving low request latency necessitates a performant database setup.

In addition to the core catalogue components, proper deployment should also include a reverse proxy/API gateway. This component should support advanced configuration features such as caching and rate limiting to optimise service performance.

Authorisation within the catalogue service itself is managed by connecting to a Keycloak instance for access token verification. However, it is important to note that the majority of requests are read-only and do not require authorisation. Additionally, the integration with an authentication system like Keycloak also facilitates the inclusion of private collections, which are accessible only to specific users.

7.4 Operational Management

For the operational deployment of the product catalogue, the APEx project is considering several configurations tailored to different project needs:

- **ESA Project Result Repository (PRR):** This instance serves as a long-term repository for preserving project results under the auspices of ESA. It is hosted on the ESA Cloud and expects the highest request volume. Unlike other instances, it is not instantiated dynamically due to its long-term nature.
- **Project-Specific Catalogue with Shared Database:** This option provides a cost-effective solution suitable for most projects, leveraging a shared database instance.
- **Project-Specific Catalogue with Isolated Database:** Designed for the most demanding projects, this configuration ensures dedicated resources but comes at a higher cost.

These different configurations allow us to offer attractive choices to projects of varying sizes and requirements. As it is challenging to predict the demand for the service, initial deployment may focus on the ESA PRR option alongside a second instance (with a shared database) for testing and storing intermediate project results.

For project-specific catalogues, continuous monitoring of the Kubernetes cluster is essential to maintain optimal performance and availability. Automated maintenance tasks such as updates and backups are routinely performed to minimise downtime and ensure data integrity.

7.5 Usage

This Product Catalogue service has been instantiated for the following projects:

Table 6. Usage of Product Catalogue

Project	Description	Status	URL
APEx	APEx has its own dedicated product catalogue and browser that are currently integrated as the Data Catalogue in the main APEx website	Done	https://catalogue.apex.esa.int/ https://browser.apex.esa.int/
APEx Demo	To showcase the APEx capabilities, a demo environment has been instantiated, including a product catalogue.	Done	https://catalogue.demo.apex.esa.int/ https://browser.demo.apex.esa.int/

World Ecosystem Extent Dynamics (WEED)	A product catalogue has been created for the WEED project to enable the project to publish results directly from an openEO workflow	Done	https://catalogue.weed.apex.esa.int/ https://browser.weed.apex.esa.int/

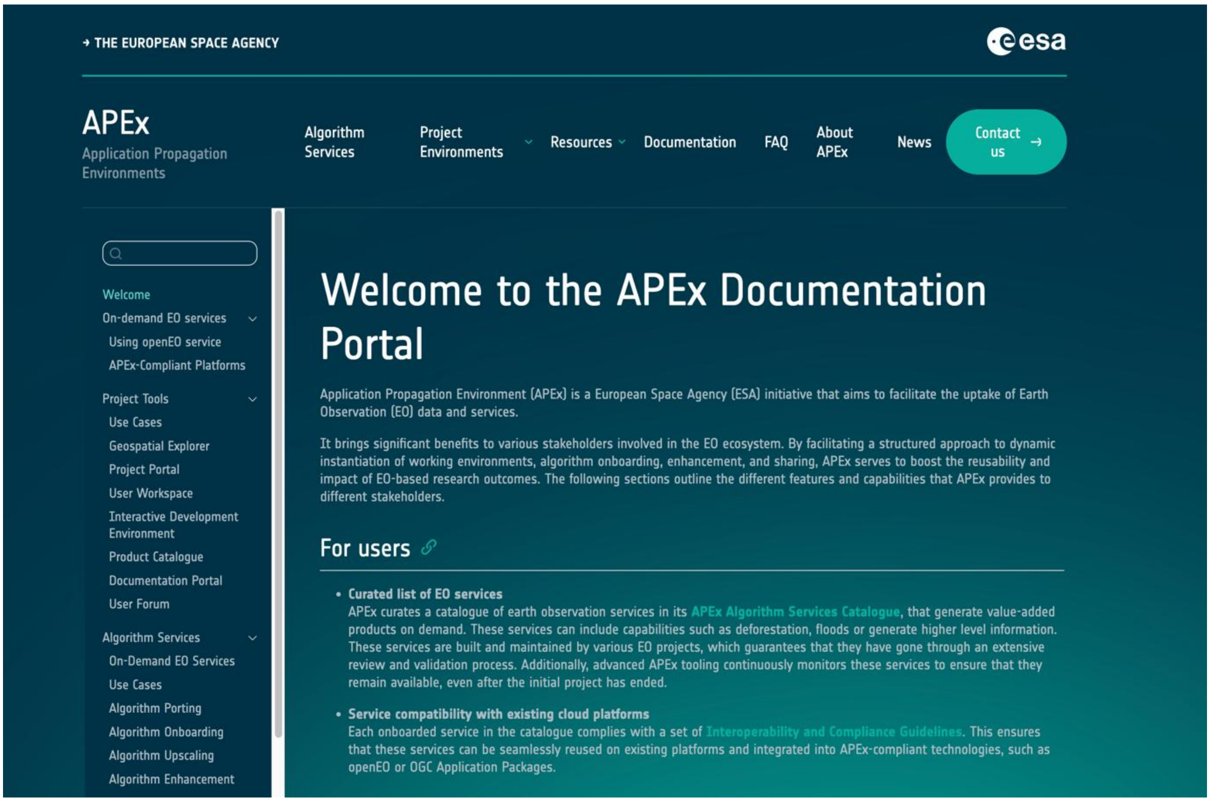
7.6 Roadmap

The project team will prioritise user feedback from the instantiated catalogues to guide the ongoing evolution of service features. In addition, there is close collaboration with the EOEPCA building blocks as they evolve, which may result in new features being integrated into the APEx service. One such feature currently under close consideration is the integration of an Admin UI.

8 DOCUMENTATION PORTAL

8.1 Overview

The documentation portal service of APEx enables projects to effortlessly create and host a documentation portal utilising the open source Quarto framework. Quarto is a comprehensive open-source system designed for scientific and technical documentation, using Markdown as the primary editing language and supporting the visualisation of both Jupyter and R notebooks. The proposed approach is complementary to hosting documents in PDF or Word format on the web portal. It offers significant advantages over these traditional formats, especially for technical writing with scientific plotting and code snippets. It also allows for introducing interactive elements in documentation. Projects that do not feel comfortable with an approach based on Markdown for editing and Git for versioning are still free to use those traditional formats.



Welcome to the APEx Documentation Portal

Application Propagation Environment (APEx) is a European Space Agency (ESA) initiative that aims to facilitate the uptake of Earth Observation (EO) data and services.

It brings significant benefits to various stakeholders involved in the EO ecosystem. By facilitating a structured approach to dynamic instantiation of working environments, algorithm onboarding, enhancement, and sharing, APEx serves to boost the reusability and impact of EO-based research outcomes. The following sections outline the different features and capabilities that APEx provides to different stakeholders.

For users

- **Curated list of EO services**
  APEx curates a catalogue of earth observation services in its APEx Algorithm Services Catalogue, that generate value-added products on demand. These services can include capabilities such as deforestation, floods or generate higher level information. These services are built and maintained by various EO projects, which guarantees that they have gone through an extensive review and validation process. Additionally, advanced APEx tooling continuously monitors these services to ensure that they remain available, even after the initial project has ended.
- **Service compatibility with existing cloud platforms**
  Each onboarded service in the catalogue complies with a set of Interoperability and Compliance Guidelines. This ensures that these services can be seamlessly reused on existing platforms and integrated into APEx-compliant technologies, such as openEO or OGC Application Packages.

Figure 1919. APEx Documentation Portal

Quarto generates fully customisable online documentation portals tailored for scientific publications, featuring interactive visualisations. It supports various scientific elements, including formulas, diagrams, and code. Additionally, Quarto can produce PDF or Word documents, making it ideal for scenarios where user documentation needs to be transformed into a project deliverable.

The content of the documentation portal is managed through either a new or existing GitHub repository. This allows portal managers and editors to edit the content in a well-known and familiar environment. Although some technical documents may be stored externally, Quarto enables referencing of these documents within the documentation portal. Additionally, thanks to the capabilities of GitHub, projects have the option to employ their own QA processes by leveraging branching, issues, and pull requests. Using GitHub Actions, the latest changes to the content will be automatically synchronised to the documentation portal.

8.2 Showcase Scenarios

A documentation portal can support several project scenarios, including:

1) **User Manuals and Guides**: Develop and store user manuals and guides for any tools or software created during the project. This ensures that users can effectively utilise the provided tools.

2) **Training and Onboarding**: Create and store training materials, tutorials, and onboarding guides for new team members or external visitors. This helps them quickly understand the project scope, tools, and processes.

3) **Technical Project Documentation**: Maintain detailed technical documentation, including software code samples, algorithm descriptions, system architectures, and hardware specifications. This is essential for sharing technical information within the project.
4) **Research and Analysis Documentation**: Document methodologies, analytical processes, and research findings. This helps ensure the reproducibility of results and provides a reference for future research.

5) **Project Planning and Management**: Store project plans, timelines, milestones, and deliverables in the documentation portal. This keeps everyone informed about project progress and deadlines.

8.3 Technical Architecture

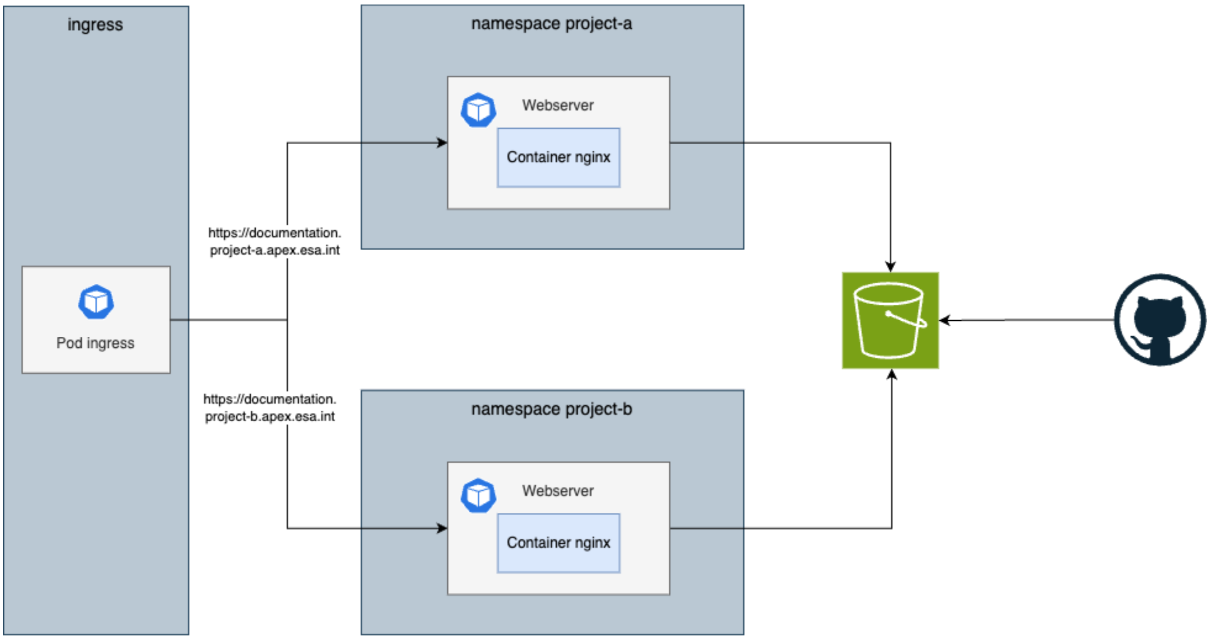


Figure 20. Technical architecture of the Documentation Portal Service

This infrastructure is designed to host the documentation portals for multiple projects, each within its own Kubernetes namespace. The setup ensures isolation, scalability, and efficient management of resources. The main components include an ingress controller, a web server and storage.

The overall architecture is defined by the following components:

- **Ingress**: This component is responsible for routing external traffic to the appropriate project namespace based on the URL. It uses rules to direct traffic to *documentation.project-x.apex.esa.int*.
- **Namespace:** Each project (e.g., *documentation.project-a.apex.esa.int and documentation.project-b.apex.esa.int*) has its own namespace for resource isolation. Each instantiated namespace will spawn the following components:
- **nginx**: This service will host the static pages that are generated by Quarto and stored on S3.
- **S3**: Provides persistent storage for the static pages that are generated by the Quarto framework.
- **GitHub**: A GitHub repository will be used to manage all content of the documentation portal. If changes are made, Quarto should generate the static pages and upload them to the S3 storage.

8.4 Operational Management

The documentation portal will primarily serve static pages, employing a straightforward approach that only requires users to have access to a web server. Given its minimal resource usage, the service deployment involves creating a web server pod with fixed resource allocation. As the project progresses, we will evaluate the necessity for increased scalability.

However, based on past experiences, a simple setup typically fulfils the requirements of most projects.

Continuous monitoring of the Kubernetes cluster is crucial for maintaining optimal performance and availability of the instantiated documentation portals. Automated maintenance tasks, including updates and backups, are regularly conducted to minimise downtime and uphold data integrity.

8.5 Usage

This Documentation Portal service has been instantiated for the following projects:

Table 7. Usage of Documentation Portal

Project	Description	Status	URL
APEx	APEx uses the Documentation Portal service to host its online documentation portal	Done	https://esa-apex.github.io/apex_documentation/

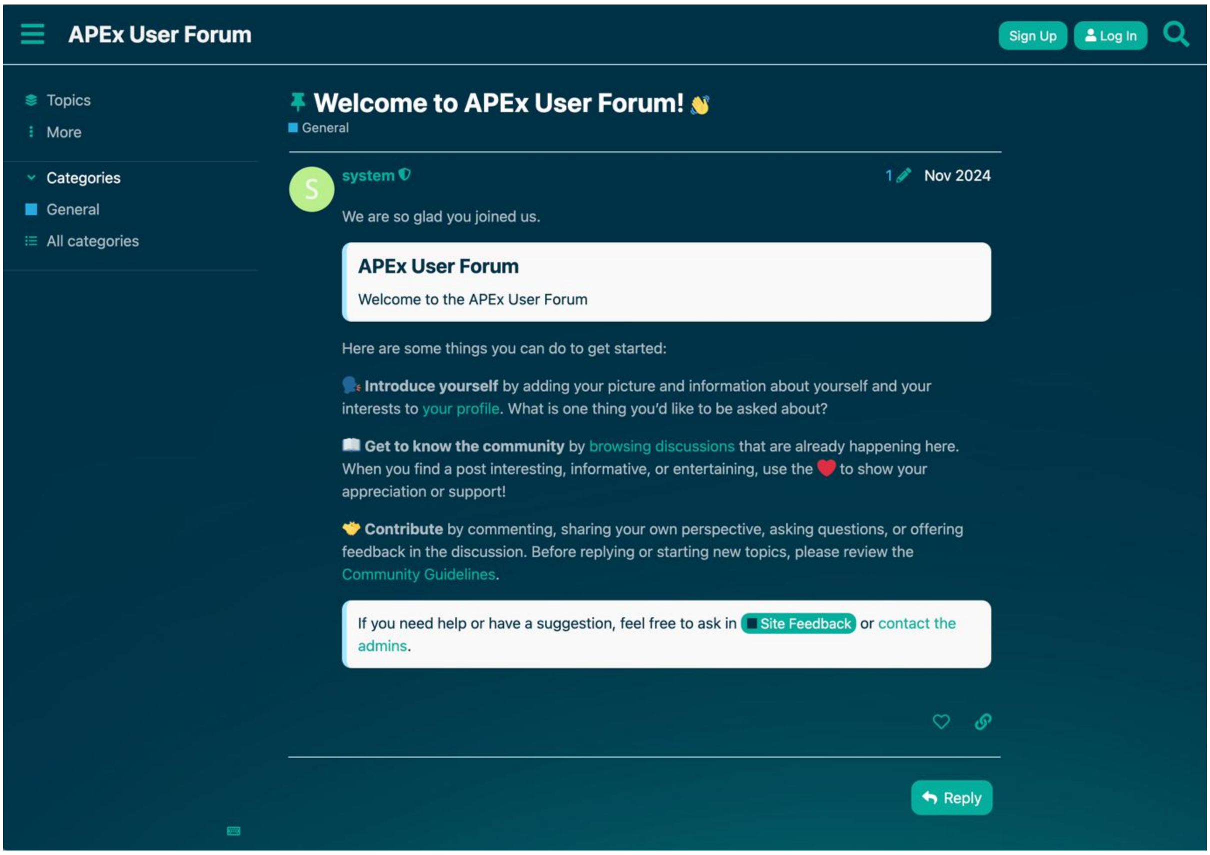
8.6 Roadmap

The roadmap for this service will evolve with future versions of the document and will be shaped by user feedback collected during its provision to projects.

9 USER FORUM

9.1 Overview

The forum service enables projects to request their own user forum on demand, providing a critical tool for community building and user support. By offering a dedicated space for discussions, the forum allows the project, its stakeholders, and users to interact seamlessly, fostering a sense of community and collaboration. This interaction not only aids in addressing user concerns and questions but also encourages the sharing of knowledge and best practices among members.



Welcome to APEx User Forum! 👋

General

system 1 Nov 2024

We are so glad you joined us.

APEx User Forum

Welcome to the APEx User Forum

Here are some things you can do to get started:

🗣️ **Introduce yourself** by adding your picture and information about yourself and your interests to your profile. What is one thing you'd like to be asked about?

📖 **Get to know the community** by browsing discussions that are already happening here. When you find a post interesting, informative, or entertaining, use the ❤️ to show your appreciation or support!

🤝 **Contribute** by commenting, sharing your own perspective, asking questions, or offering feedback in the discussion. Before replying or starting new topics, please review the Community Guidelines.

If you need help or have a suggestion, feel free to ask in Site Feedback or contact the admins.

Figure 2121. APEx User Forum

The underlying technology for the forum service is Discourse, a widely used open-source forum software renowned for its robustness and flexibility. Discourse has been successfully implemented in various ESA projects, demonstrating its reliability and effectiveness. One of the standout features of Discourse is its comprehensive admin interface, which empowers projects to customise the user forum to meet their specific requirements. This customisation can include modifying the forum's appearance, setting user permissions, and managing content. Moreover, Discourse supports a wide range of plugins that can significantly enhance the forum's functionality, providing a versatile solution that can adapt to the evolving needs of any project.

9.2 Showcase Scenarios

A user forum can support several project scenarios, including:

- **Knowledge Sharing and Collaboration**: Forums bring together experts, researchers, and enthusiasts to share knowledge and insights, fostering a collaborative environment. Users can post questions and receive answers from the community, helping to solve problems more efficiently.

- **Community Building**: Forums help build a sense of community among users who share a common interest in the project. Users can connect with others, potentially leading to additional collaborations and partnerships.

- **Feedback and Improvement**: Project members can receive direct feedback from users, allowing for continuous improvement and refinement of the project tools and methodologies. Users can suggest new features or enhancements, ensuring the project evolves according to user needs.

- **Support and Training**: Forums provide a platform for users to seek technical support and troubleshooting assistance. Users can participate in discussions to enhance their skills and knowledge.

- **Awareness and Outreach:** Forums can be used to disseminate information about project updates, events, and relevant news.

9.3 Technical Architecture

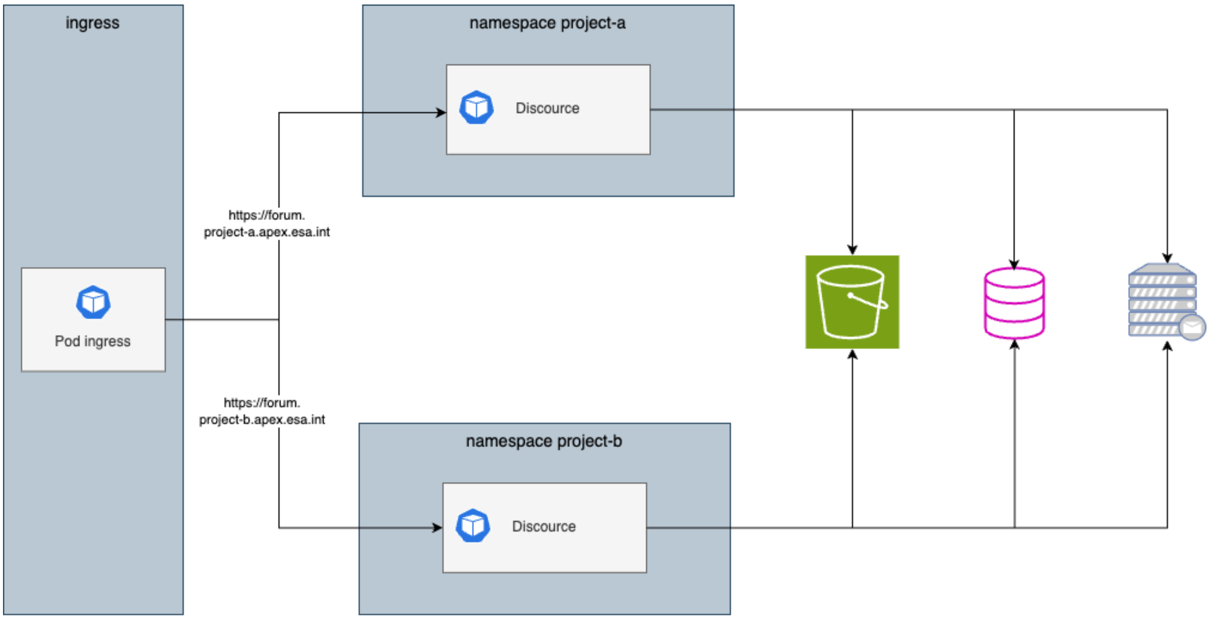


Figure 22. Technical architecture of the User Forum Service

This infrastructure is designed to host the user forums for multiple projects, each within its own Kubernetes namespace. The setup ensures isolation, scalability, and efficient management of resources. The main components include an ingress controller, the Discourse forum, storage, a database and a mail server.

The overall architecture is defined by the following components:

- **Ingress**: This component is responsible for routing external traffic to the appropriate project namespace based on the URL. It uses rules to direct traffic to *forum.project-x.apex.esa.int*.
- **Namespace**: Each project (e.g., *forum.project-a.apex.esa.int* and *forum.project-b.apex.esa.int*) has its own namespace for resource isolation. Each instantiated namespace will spawn the following components:
    - **Discourse**: This service will host the actual Discourse forum.
- **Storage:** Provides persistent storage for the user forum assets.
- **Database Managed Service**: This is a centralised managed database service that is linked to the Discourse instances. By default, the database service will be shared among all user forum instances to reduce management overhead and minimise costs. Despite the shared infrastructure, each forum will remain completely isolated to ensure project separation. For highly active user forums, APEx will also offer the option to host the forum on a dedicated database infrastructure, though this option comes at a higher cost.
- **Mail server**: An email SMTP server that allows the user forum to send email notifications.

9.4 Operational Management

The user forum operates within a Kubernetes environment shared across all APEx Instantiation Services. A dedicated namespace will be used to host the user forum that was requested for a single project. An important consideration in this setup is the cost of the managed database service. As the cost of a database can be high considering the traffic it needs to handle for a specific project, APEx is envisioning to provide two tiers to support projects of different scales:

- A basic offering, where the user forums are sharing a database instance, can support many low-traffic forums at a lower cost.
- A large forum offering with increased resource limits and its own dedicated database instance. This option mainly targets larger projects with a larger user base and high forum usage but comes at a higher cost.

Continuous monitoring of the Kubernetes cluster is crucial for maintaining optimal performance and availability of the instantiated user forums. Automated maintenance tasks, including updates and backups, are regularly conducted to minimise downtime and uphold data integrity.

9.5 Usage

This User Forum service has been instantiated for the following projects:

Table 8. Usage of User Forum

Project	Description	Status	URL
APEx	APEx uses the User Forum service to host its online user forum	Done	https://forum.apex.esa.int/

9.6 Roadmap

The roadmap for this service will evolve with future versions of the document and will be shaped by user feedback collected during its provision to projects.

vision on technology
for a better world

vito